Identifying Prosodic Prominence Patterns for English Text-to-Speech Synthesis

Leonardo Badino

Doctor of Philosophy

Institute for Communicating and Collaborative Systems

School of Informatics

University of Edinburgh

2010

Abstract

This thesis proposes to improve and enrich the expressiveness of English Text-to-Speech (TTS) synthesis by identifying and generating natural patterns of prosodic prominence.

In most state-of-the-art TTS systems the prediction from text of prosodic prominence relations between words in an utterance relies on features that very loosely account for the combined effects of syntax, semantics, word informativeness and salience, on prosodic prominence.

To improve prosodic prominence prediction we first follow up the classic approach in which prosodic prominence patterns are flattened into binary sequences of pitch accented and pitch unaccented words. We propose and motivate statistic and syntactic dependency based features that are complementary to the most predictive features proposed in previous works on automatic pitch accent prediction and show their utility on both read and spontaneous speech.

Different accentuation patterns can be associated to the same sentence. Such variability rises the question on how evaluating pitch accent predictors when more patterns are allowed. We carry out a study on prosodic symbols variability on a speech corpus where different speakers read the same text and propose an information-theoretic definition of optionality of symbolic prosodic events that leads to a novel evaluation metric in which prosodic variability is incorporated as a factor affecting prediction accuracy. We additionally propose a method to take advantage of the optionality of prosodic events in unit-selection speech synthesis.

To better account for the tight links between the prosodic prominence of a word and the discourse/sentence context, part of this thesis goes beyond the accent/no-accent dichotomy and is devoted to a novel task, the automatic detection of contrast, where contrast is meant as a (Information Structure's) relation that ties two words that explicitly contrast with each other. This task is mainly motivated by the fact that contrastive words tend to be prosodically marked with particularly prominent pitch accents.

The identification of contrastive word pairs is achieved by combining lexical information, syntactic information (which mainly aims to identify the syntactic parallelism that often activates contrast) and semantic information (mainly drawn from the WordNet semantic lexicon), within a Support Vector Machines classifier.

Once we have identified patterns of prosodic prominence we propose methods to incorporate such information in TTS synthesis and test its impact on synthetic speech naturalness trough some large scale perceptual experiments.

The results of these experiments cast some doubts on the utility of a simple accent/no-accent distinction in Hidden Markov Model based speech synthesis while highlight the importance of contrastive accents.

Acknowledgements

Many thanks to Rob Clark, his assistance, support and the calm he brought to me have been what I really needed.
Thanks to Mark Steedman for his valuable feedback and for his work which has been a great source of inspiration.
Thanks to Sasha Calhoun for helping me to better understand the relationship between information structure and prosodic prominence.
Thanks to all people in CSTR, I have learnt so much from the CSTR meetings and I enjoyed all the Friday nights at the pub spent with them. A special thank to Sebastian for helping me to build the HMM based voices and for many other things.
Thanks to my Italian (and pseudo-Italian) fellas, Moreno, Daniele, Erida, Dario and Elena for making me feel much less homesick than I would have been without them.
Thanks to Nicola for lending me the flat where I wrote the last part of this thesis.
A huge thanks to my mother, my sister and my parents-in-law for their unconditional support and generosity.
A big thanks to my son, Samuele, for all the happiness he has brought in my life (and for having showing me that I do not need to waste eight hours per day in sleeping to get a good life).
A huge thanks to my father, the memory of him and the strength with which he always faced any difficulty give a sense to everything I do.
Finally the biggest thanks to Laura, for her love.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Leonardo Badino*)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis addresses the problem of the identification of natural patterns of prosodic prominence and their generation in Text-to-Speech (TTS) synthesis.

In state-of-the-art TTS systems prosodic prominence patterns are predicted from text by relying on features that very loosely account for the combined effects of syntax, semantics, word informativeness and salience on prosodic prominence.

Such a simplification can lead to a mismatch between the semantic, pragmatic and syntactic content of an utterance and its prosodic realization. As a consequence it may not only affect the naturalness of the synthetic speech but also cause misunderstandings in human-computer spoken communication. Consider the following possible dialogue excerpt from a flight-planning application:

(1.1) User: I'd like a flight tomorrow morning to Edinburgh.
System: I have two flights tomorrow evening with British Airways.

Without an appropriate prosodic pattern emphasizing the word "evening" the user infers that "morning" has been misrecognized as "evening" and (in the most optimistic case) will try to repair the misunderstanding by rephrasing her request.

It is commonly held within the speech synthesis community that the synthetic generation of prosodic prominence (and more in general of prosody) tends to sound inadequate in long utterances and/or when utterances are in context, making TTS synthesis not yet satisfactory in some applications like book reading and automated spoken dialogues.

Despite that this problem has received little attention, attention that has been focused on only one aspect of the problem. In fact TTS researchers addressing this problem are concerned about the prosodic rendering of the pragmatic and semantic content of an utterance but do not address the problem of processing that content, delegating such task to the developer of an application using the TTS system (for example through a markup language) or to an "artificial mind" of which the TTS system is relegated to a tool to communicate with the external world.

The main goal of this thesis is to improve the prediction from text of prosodic prominence patterns. According to phonological accounts on prosody, the prediction of prosodic prominence patterns is approximated to the prediction of sequences of pitch accents, i.e. symbolic events associated to pitch movements over the most prominent (actually perceived as most prominent) syllables in an utterance and correlated with increased phone duration and intensity, better voice quality and other spectral effects.

In English (and some other languages) pitch accenting is mainly determined by the

degree of informativeness and salience of words which in turns is affected by the context (both discourse context and sentence-internal context) and by intrinsic properties of words (e.g., properties indicating if a word is a content or a function word).

The main context related mechanism that controls the distribution of the salience of information in an utterance is generally known as *focus marking*. *Focus marking* distinguishes the words of the utterance that are presupposed (usually called *background*) from the words that "contribute distinguishing the actual content of the utterance from the alternatives the context makes available" usually referred to as *focus* or *kontrast*[1].

Typical examples used to illustrate *focus marking* are question-answer pair examples like the following:

(1.2) Q: Which football team does Paul support?
A: Paul supports Arsenal.

where "Paul supports" is the presupposed part of the utterance whereas "Arsenal" is the focused item.

The focused item of a sentence can be seen as an item evoking (i.e., bringing to attention) a set of alternatives (to the focused item) made available by the context. So in example (1.2) the word "Arsenal" evokes the set of football teams[2].

When the set of evoked alternatives is limited to few words or just one word the focused item often appears to be clearly contrastive[3]). For example in (1.1) the word "evening" evokes and contrasts with the word "morning".

Although it is under debate whether this specific case of *focus* (which may be referred to as *restricted* or *closed focus*) deserves its own semantic category several empirical studies have shown phonological and acoustic differences (depending on the perspective one looks at such distinction) between its prosodic marking and that of accented but not contrastive items. In general a *contrastive accent* is usually more prominent than a "standard" accent and it is occasionally referred to as "emphatic accent".

The phonetic properties of *contrastive* accents make the identification of contrastive items very attractive for TTS applications. Such task would allow to go beyond a simple distinction between accented and unaccented words and move towards more

[1] Here the term *kontrast* has the same meaning of *focus* and should not be confused with the concept of contrast (although we will see in chapter 2 that *focus* and contrast may be seen as two very similar if not identical concepts.)

[2] Or British football teams or London's football teams. The set evoked need not to be well delimited

[3] We will see in chapter 2 that actually *focus* and contrast may be seen as one single concept even when the set of alternatives evoked by the focused item is not limited to a close small set (like in example (1.2).

expressive TTS systems.

Under this perspective, part of this thesis (chapter 5) is devoted to the automatic labelling of *contrast*, where *contrast* is meant as a relation that ties two words that explicitly contrast with each other or two words where when one word contrasts with the other (as in example 1.1).

The approach proposed to accomplish this novel task is mainly based on the observation that contrastive elements are "similar in a way but different in another", where similarity is not only semantic similarity but also syntactic similarity (e.g. words sharing the same syntactic function or/and modifying the same syntactic head). A combination of simple lexical features, syntactic features (mainly in the form of syntactic dependencies) and semantic information (mainly in the form of semantic relations drawn from the WordNet semantic lexicon, e.g., hypernym, antonym, etc...) are combined into a Support Vector Machines (SVM) based *contrast* tagger.

Focus marking is not the only mechanism that controls the salience and organization of information and affects pitch accenting. Usually when only one word is focused (*narrow focus*) that word carries the most prominent accent, but that does not mean that words in the *background* can not be accented. Moreover when the focused item is a phrase or a sentence (*broad focus*) then pitch accentuation is determined by other factors.

One of these factors, another discourse related factor which is actually somehow intertwined with *focus marking*, is the *new/given* dichotomy with *new* words tending to be accented and *given* words (i.e., words that have already been, explicitly or implicitly, mentioned in the discourse) tending to be deaccented.

Non-discourse related factors that affect pitch accenting are mainly features that indicate the intrinsic informativeness of words. Content words are usually more informative (i.e., convey more unexpected information) than function words and so are more likely to be accented. Similarly, rare words are more likely to be accented than frequent words.

In automatic pitch accent prediction the use of discourse related features have been very limited, perhaps because an accurate extraction of these features seems impracticable and when approximated solutions to extract them have been used the impact of these features on accent prediction has been marginal.

In chapter 3, where ("standard") pitch accent prediction is addressed, there is no ambition to accurately extract discourse related features and the attempt of outperform-

ing state-of-the-art accent prediction is mainly based on the extraction of new features that aim to: 1) be complementary to the statistical features proposed in previous work to estimate the informativeness of words (i.e., probability of seeing a word), of the informativeness of the concepts conveyed by the words, and 2) better account for the effects of syntax on pitch accenting.

Another factor affecting the accentuation value of a word (i.e., ±accent) is the "accentuation history", i.e., the accentuation values of the preceding words. To account for the "accentuation history" some previous studies have proposed machine learning techniques such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) as opposed to more traditional (in the literature on pitch accent prediction) techniques like Classification Trees in which the placement of a pitch accent is assumed to be independent of the accentuation history. Since the literature seems to lack of a fair comparison between the two approaches part of chapter 3 is devoted to such comparison.

All factors affecting pitch accentuation act in a non-deterministic way, meaning that, given a sentence, different accentuation patterns are equally acceptable and all convey the same meaning. Such variability, which is evident when comparing the accentuation patterns of different speakers that read the same text or even when comparing the patterns of a single speaker that reads the same text at different times, is due to the optionality of (some) accents. Consider the example:

(1.3) Q: What did Arun's mother think?
A: Arun's mother disapproved.

an accent on "disapproved" is compulsory (in a natural-sounding utterance) because "disapproved" is focused. However, because of rhythmic effects, an accent on "mother" would be perfectly acceptable (as a non-accent would be), provided that the accent "mother" sounds less prominent than the accent on "disapproved". So the accent on "mother" is optional while the accent on "disapproved" is compulsory (and "meaningful").

Prosodic variability raises issues on the correct evaluation of pitch accent predictors. Simply marking as correct a predicted accent value when it matches the accentuation value in the test data becomes an incorrect evaluation since a predicted accent sequence that is different from the test sequence may be as correct as the test sequence.

An alternative evaluation metric that accounts for variability in pitch accent placement is proposed in chapter 4. The metric is based on an information-theoretic definition of optionality that sees optionality as a continuous-valued variable. This definition

of optionality leads to a more correct evaluation of accent prediction with respect to evaluation metrics proposed in previous work and based on a binary definition of optionality (i.e., ±optional).

Once we have worked on the identification of prosodic prominence patterns (both in form of standard and contrastive pitch accent sequences) the next natural step is that of validating the utility of such work in the generation of prosodic prominence in TTS synthesis. That is the goal of chapter 6, where, through large scale perceptual tests several questions on the utility of accent prediction in TTS synthesis are addressed. The main questions are: is a substantial improved pitch accent prediction reflected in better prosody generation? Can a TTS system benefit from the knowledge of the degree of optionality of accents? Can we generate accurate contrastive accents in HMM-based speech synthesis?

1.1 Scope of the thesis

Since the target of improving the identification and generation of natural patterns of prosodic prominence for TTS synthesis requires several different issues to be addressed, ranging from text analysis to acoustic modeling of prosody, which we do not think can be all equally covered in a PhD thesis work, this thesis addresses the issues related to text analysis more deeply than those related to speech synthesis and consequently is more focused on the linguistic features that affect prosodic prominence than on issues related to the acoustic correlates of prosody.

The motivation behind this bias stems from the need to counterbalance the bias of previous work on the speech synthesis part. In fact almost all previous studies on prosodic prominence modeling (and more in general on prosody modeling) for TTS propose solutions to improve the prosodic quality of synthetic speech that "only" concern speech signal and data. For example solutions to collect more expressive speech data, or solutions to generate more natural F0 curves given linguistic specifications. The linguistic information used is usually very simple (e.g. Part-of-Speech, position of the word in the sentence, position of the syllable in the word, etc...) and independent of the syntactic and semantic context, or, when it is more complex and context-dependent (usually represented through prosodic symbols from the ToBI symbols set), its identification is left to an "artificial mind" or to the TTS user. This lack of a deep text analysis is acceptable when the TTS system is required to read out-of-context sen-

tences or when it is integrated within a dialogue system (usually working on limited domains, e.g. flight booking) that provides all the necessary contextual information to the TTS system.

However there are still several applications, usually more challenging open-domain applications, where a TTS system can only rely on itself, applications where there are no external agents or, where there is an external agent but it does not provide to the TTS systems all the information it needs to generate an appropriate prosodic pattern.

1.2 Thesis Outline

The following is a chapter by chapter outline of this thesis.

Chapter 2: this is the background chapter where we will have a closer look to the concepts of prosodic prominence, pitch accenting, focus and contrast, to the main factors affecting pitch accenting and to the previous work on (both "standard" and "contrastive") pitch accent labeling. A description of the two main speech synthesis techniques (unit selection and Hidden Markov Model (HMM) based speech synthesis) and a review of previous work on prosodic prominence generation for TTS synthesis conclude the chapter.

Chapter 3: this chapter concerns the automatic labeling of pitch accents. New statistical and syntactic features are proposed as features complementary to the most predictive features from previous work and their utility is tested on both read and spontaneous speech. Different machine learning techniques are compared both theoretically and empirically to investigate which techniques (and which of their properties) are most suitable for pitch accent prediction.

Chapter 4: the first part of this chapter presents a study on the variability of pitch accent placement in read speech. The following part introduces and motivates an information-theoretic measure of the optionality of prosodic symbols. The chapter concludes with the description of a novel method that takes advantage of the optionality of pitch accents (and potentially of other prosodic categorical events) to improve the speech quality of unit selection synthesis.

Chapter 5: this chapter concerns the automatic labeling of *contrast*. Examples of *contrast* from a corpus of spontaneous spoken English are shown in order to give a better definition of the task and the difficulties involved, and to introduce and motivate a novel Support Vector Machine (SVM) based *contrast* tagger. After an analysis of error of the tagger, new methods are proposed to improve the tagger accuracy. One of

the main problem in *contrast* tagging is the lack of training data. The chapter concludes with the description of the implementation of an algorithm of Active Learning SVM for *contrast* labelling whose goal is a fast creation of manually labelled training data sets.

Chapter 6: in this chapter the impact of most of the work discussed in chapter 3, 4 and 5 on TTS synthesis (both unit selection and HMM based) is investigated through a series of large scale perceptual tests. All the test results are preceded by a description of how the information provided by the pitch accent predictor and *contrast* tagger are integrated in a TTS system and what solutions increase the utility of such information.

Chapter 7: this chapter concludes the thesis with a summary of the main results achieved in this thesis, the main problems encountered, the problems that came up from the findings of this thesis, possible solutions and future directions.

Chapter 2

Background

2.1 Prosodic Prominence and Pitch Accents

Although this thesis is not firmly anchored to any linguistic theory of intonation, it is however based on some fundamental assumptions of the Autosegmental-Metrical (AM) theory of Intonation[1] (based on Liberman (1975), Bruce (1997) and Pierrehumbert (1980)), the first (i.e., most fundamental one) of which says that the pitch[2] contour consists of a string of phonological *pitch events* interspersed with phonologically underspecified *transitions*. *Pitch events* are divided in *pitch accents*, which are variations of pitch associated with prominent syllables, and *edge tones*, which are segments of pitch associated with prosodic boundaries.

Another basic assumption of the AM theory is that pitch accents are "post-lexical" events, meaning that their placement and type is not determined at the lexical level (like lexical stress) but at a higher level.

Further, AM theory makes a clear distinction between pitch accents and stress, where stress is not simply lexical stress but a more "abstract" concept. According to AM theory every utterance has a stress pattern that "reflects a set of prominence relations between the elements of the utterance" (Ladd (1996)). The stress pattern is organised in a binary tree structure where two siblings are always tied by a weak-strong relation that states which of the two is most prominent. Figure 2.1 shows an example of relative prominence tree for the noun phrase "labor union" where the lexical stress on "labor" is stronger (i.e., perceived as more prominent) than the lexical stress on "union"[3]. The same kind of tree can be built for a whole intonational phrase (i.e., a part of the utterance delimited by "strong" prosodic breaks).

This hierarchical representation of prosodic prominence assumes that in English several degrees of prominence can be perceived.

However it is not clear whether different levels in the hierarchy have different phonetic correlates. What is clear is that, save some exceptions, in spoken English prominent syllables (i.e., the "strong" siblings, at any level of the tree) can be marked by a

[1] Throughout this thesis we will consider intonation as the linguistic component of prosody dealing with F0 (or pitch, see next footnote), which is an acoustic attribute of prosody. Other acoustic attributes of prosody are duration and intensity.

[2] Actually we should say F0 instead of pitch, being the former an acoustic property of speech and the latter its psycho-physical correlate, i.e., the perceived (by humans) F0. However, throughout this thesis we will use the two terms interchangeably in accord with most of the literature.

[3] Note that the weak-strong relation is not defined for any couple of items. For instance in the example in figure 2.1 the prominence relation between the last syllables of the two words is not defined, both are subordinated to the strong branch but neither of the two is subordinate to the other. When uttering "labor union" the last syllable of "labor" could be more salient than the last last syllable of "union", or vice-versa, without violating the prominence tree which can not account for such difference.

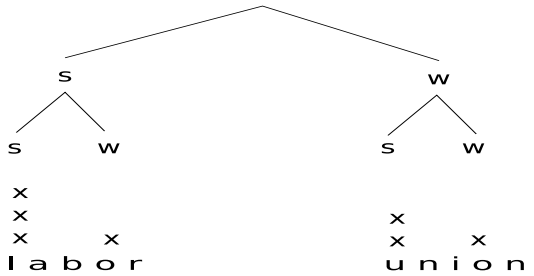

Figure 2.1: *Prominence tree of "labor union". The letter 's' and 'w' stand for relatively strong and relatively weak sibling respectively. The prominence tree can be mapped into a prominence grid in which the number of 'x' indicates the level of prominence of a syllable. For example, the first syllable of labor has the higher number of 'x' meaning that it is the strongest syllable. Note that the prominence tree is always a binary tree independently of the number of items in the intonational phrase.*

pitch accent while "weak" syllables can not. At least in English the most prominent syllable in a phrase must be accented, so citing Beckman and Pierrehumbert (1986) "[accents] are some sort of culmination of prominence". As a consequence pitch accents serve as cues of prominence, and can be see as the strongest cue of prominence, but are not necessarily the only cue of prominence (at the post-lexical level).

The prominence tree also implies a hierarchy of pitch accents that goes beyond the binary $\pm$accent value, with pitch accents occurring on the strongest words being more prominent than those on weaker words.

The default weak-strong relation of the minimum sub-tree (i.e., the tree only having one parent node and two children), which biases a stronger prominence on the right branches, also accounts for the fact that in English the rightmost accent (usually referred to as nuclear accent or primary accent) in an prosodic phrase is usually the accent perceived as most prominent. As we will see in section 2.2.1 the nuclear accent is on the pragmatics side the most important accent since it marks focused words, i.e., the most salient words given the discourse context.

H*	simple high	(canonical declarative)
L*	simple low	(yes-no question)
L+H*	rising to high from low	(contrastive focus)
L*+H	"scooped" late rise	(pragmatic uncertainty)
H+!H*	fall onto stress	(pragmatic inference)

Figure 2.2: *ToBI pitch accents. (from Pitrelli et al. (1994))*

In the AM theory a taxonomy of pitch accents, and of edge tones (divided into boundary and phrase tones), is defined according to the primitive level tones that compose them. The primitive level tones (also referred to as pitch targets) are High(H) and Low(L).

The most up-to-date taxonomy of pitch accents stemming from the AM theory is described in the ToBI (TOnes and Break Indices) annotation system (Silverman et al. (1992) and Pitrelli et al. (1994)). In ToBI five types of accent are defined: H*, L*, L+H*, L*+H and H+!H* (see figure 2.2). The '*' symbol after a level tone means that the tone (starred tone) is the central tone of the accent. In some accent types, starred tones can be preceded or followed by a "leading" or a "trailing" tone respectively. The diacritic "!" indicates that the following tone is down-stepped, that is the tone is lowered because of a compression of the pitch range.

No use of the ToBI pitch accent taxonomy will be made in this thesis and ToBI accent types will be only mentioned when describing previous work in which such taxonomy is used. The work described in the following chapters is mainly concerned with the placements of pitch accents (whatever the ToBI type) and only distinguishes between "standard" pitch accents (which can be roughly associated to H* accents) and "contrastive" pitch accents (which are usually associated to L+H* accents, see Pitrelli et al. (1994)).

The main motivation behind this simplification is that one of the main goals of this thesis is to identify sequences of pitch accents from text and we do not believe that an accurate prediction of all the accent types in a sentence can be achieved by looking at text only and, above all, "simpler" problems, like the prediction of $\pm$ accents or the distinction between nuclear and non-nuclear accents, have not been completely solved yet or have still to be solved.

Moreover, note that H* and L+H* are usually by far the most frequent accent types. For example in the Boston University Radio News corpus (BURNC, Ostendorf et al.

(1995)), which is annotated according to ToBI conventions, H* and L+H* accents make up 94% of all accents (reported by Taylor (2000)).

Since looking at pitch accent placements only is equivalent to collapsing all accent types in one unspecified accent type we may wonder what kind of useful (for TTS purposes, at least) information we can still obtain from such unspecified accents. If all the information we can extract from accents were contained in their intonational "shape" (i.e., type) the answer will be "no information". Luckily some information is still left since the phonetic correlates do not reside in F0 only. Indeed experimental evidence shows that syllables (independently of the accent type) are usually marked with greater duration (see Turk (1999) for example, where also the "scope" of (contrastive) accents is investigated). Also other phonetic cues have been associated with accents, e.g., increased intensity, better vowel quality and effects on the spectral tilt (with more energy at high frequencies, Campbell and Beckman (1997)). What seems to emerge from experimental work is that increased duration, intensity, and other acoustic correlates of pitch accents are not exclusive correlates of pitch accents but are all cues of different levels of prominence[4](i.e., the levels defined in the prominence tree mentioned above).

A further reason that justifies the importance of locating pitch accent placement is that, as we will see in section 2.6.2, in HMM-based speech synthesis it is crucial to know which linguistic items affect the velocity of change of the acoustic parameters (including F0), and accents definitely affect the velocity of change of F0 in that they are by definition associated with (phonological) pitch movements.

2.2 Pitch Accents in Context

Since a goal of this thesis is that of predicting pitch accent (both "standard" and "contrastive") placements, the main question we address is: what are the factors that affect pitch accent placement and the placement of the most prominent accents?

In several languages, including English and Dutch, there is a generally accepted direct relation between prosodic prominence and salient and informative information in an utterance, being salient and informative words usually marked with pitch accents. Note that here we make a distinction between salient and informative information where salient information is information that is relevant for the listener while informative information is information that adds new and unexpected content to the

[4]Levels that in a $\pm$accent discrimination are roughly collapsed to one of the two groups (+accent or -accent)

discourse but is not necessarily relevant for the listener.

The parts in an utterance that carry salient and informative information are determined by the discourse context, and (as we will see in section 2.3), especially when the utterance is out-of-context, by intrinsic properties of words and utterance-internal relations among words.

This section concerns the effects of discourse context on pitch accenting by looking at the relationship between pitch accenting and Information Structure. Information Structure is meant here as describing the distribution of salience and organization of information of the semantic content of an utterance in relation to the discourse context.

2.2.1 Focus and Pitch Accents

The main mechanism controlling for the organization of salience and informativeness in a sentence is the *focus/background* distinction (see section 2.2.3.1 for a formal definition). The best way to illustrate the *focus/background* is showing examples of question-answer pairs like the following:

(2.1) Q: What did Sam have for lunch?
A: Sam had RAVIOLI.

(2.2) Q: Who had ravioli for lunch?
A: SAM had ravioli.

where "ravioli" in (2.1) and "Sam" in (2.2) are focused as they are clearly the salient and informative words in the answer (i.e., they are the "actual answer within the answer"), given the question. And since they are focused then they are also accented[5]. The intuition that pitch accents are used to mark focused words (as well as new words) dates back to the 1950s (e.g., Bolinger (1958)).

The non-focused part of the answer is referred to as *background*. A more formal definition of focus and background (although we will see that there is not an unanimously agreed definition of focus) will be given in section 2.2.3.1, for the time being we will use an intuitive definition of focus in which focus is the most salient part of an utterance given the discourse context.

Focus does not imply that the focused word is the only accented word in the sentence, but implies that the accent on the focused word is the one perceived as most

[5]The implication "focused → accented" is, however, not always true. Ladd (1996) shows examples of focus without accents when focus is on a word of a prepositional phrase with pronoun or adverb objects, such as "for him" and "in there".

prominent and whose presence is the most necessary (see Ladd (2009), section 7.2.1), i.e.,the accent on a focused word is the primary accent[6].

Pitch accents on the background words may be required by rhythmic constraints. If we slightly modify example (2.1) into the following example:

(2.3) Q: What did Sam's mother-in-law have for lunch?
A: Sam's mother-in-law had RAVIOLI.

the subject "Sam's mother-in-law" is long enough to require at least an accent on "mother", although "mother" is neither informative nor salient. However the accent on "mother" has to be less prominent than the accent on "ravioli". If the prominence relation between "mother" and "ravioli" were reversed the interpretation of focus would be no more consistent with the question as the greatest prominence on "mother" would imply that the issue under discussion is who had ravioli for lunch (i.e., "Sam's mother", contrasting with some other relative or friend of Sam) instead of what "Sam's mother-in-law" had for lunch.

All previous examples are examples of *narrow focus* in that one single word ("ravioli" and "Sam" respectively) is marked by focus. When focus marks a phrase or a whole sentence the focus is generally referred to as *broad focus*. In the following example:

(2.4) Q: What happened?
A: Sam ate ravioli.

the whole answer is focused. In the presence of broad focus discourse context is no longer sufficient to determine which word carries the most prominent accent and other factors must be taken into account.

2.2.2 New/Given and Pitch Accents

The *given/new* dichotomy is another factor controlling the organization of information in relation to discourse context (see Chafe (1994), for example). In all previous examples of narrow focus ((2.1),(2.2) and (2.3)), narrow focus coincides with new information. In fact what is new in the discourse is often focused (and consequently

[6]When two or more single words are focused in the same utterance each focused word should still bear a primary accent if each of them belongs to a separate intonational phrase, which is often the case. We are not aware on any study in which the prominence relations among focused words belonging to the same utterance are investigated. However the focused words can be marked by different types of pitch accents if they are focused by different "types" of focus (e.g., thematic vs. rhematic focus, see section 2.2.3)

accented) and what is given (i.e., previously mentioned in the discourse) is often part of the background.

However focus and new information can not be collapsed to one single concept, but have to be kept separated to account for cases of focus on given information, and of new information without focus. The same applies to background and given information. In the following example:

(2.5) S1: Yesterday, Sam and Paul went to the restaurant for lunch
S2 : What did they eat?
S3: Sam had ravioli,...
S4: ...while Paul had risotto.

in clause S3 given information ("Sam") is focused. In fact both "ravioli" and "Sam" are focused (as well as "Paul" and "risotto" in S4). "Ravioli" is focused as it is new and relevant (i.e., it is a "part of the answer" that answers the question) information, while "Sam" is focused as it is relevant, i.e., it is "a part of the answer" that answers the question (it's "Sam" who had ravioli, not "Paul"), despite the fact that it is has already been mentioned in the dialogue[7].

On the other hand new information may not be focused as in the example below where the answer is "over-informative":

(2.6) S1: Yesterday, Sam went to the restaurant for lunch
S2 : What did he eat?
S3: Sam was tempted by the deluxe double cheeseburger but in the end he had ravioli.

"Ravioli" is again new and focused, while the verb phrase "deluxe double cheeseburger" is new but not focused. It can be argued that "deluxe double cheeseburger" is actually focused but the higher relevance of "ravioli" must be acknowledged.

Even agreeing with the theory of the existence of nested foci (Neeleman and Szendroi (2004) and Fery and Samek-Lodovici (2006)), that would assume "was tempted by the deluxe double cheeseburger but in the end he had ravioli" to be focused and "ravioli" to be nested focus within it, the highest relevance on the focus on "ravioli" must be acknowledged since "ravioli" is the actual "answer within the answer".

The information status , i.e., the status indicating if a word is new or given, has

[7] In section 2.2.3.1 we will see that in the two-dimensional approach to Information Structure (Vallduví and Vilkuna (1998) and Steedman (2000)) which maps Information Structure on a space defined by the two variables "theme/rheme" and "focus/background", the foci on "Sam" and "ravioli" are distinguished so that the focus on "Sam" is a thematic focus while the focus on "ravioli" is a rhematic focus.

been claimed to be prosodically marked, with new entities tending to be accented and given entities tending to be deaccented.

However the relation between information status and pitch accenting seems to closely depend on what is meant with given and new. If information status is defined in terms of word occurrences, where the first occurrence of a word is assumed to convey new information, while the following occurrences convey given information, the relation with accenting is weak (Ross and Ostendorf (1996) and Terken and Hirschberg (1994)). Terken and Hirschberg (1994) shows that simple prior mention of a word may not suffice to motivate deaccentuation and that deaccentuation is more frequent when the syntactic role (e.g., subject, object, etc...) and the surface position of the entity are both repeated. Needham (1990) shows that a word that refers to a part of a previously mentioned entity can be deaccented but only if that part is central to the object.

These studies suggest that a simple dichotomy given/new does not suffice to explain accentuation and deaccentuation, and in reality different degrees of givenness exist. For example Chafe (1994) defines a third information state which indicates whether an entity is accessible/predictable from the discourse context. This *accessible* state is located in between new and given. An accessible entity is not given because it has not been previously mentioned, but is not new either because it is accessible from the discourse context. Following Chafe's definition of information status Baumann and Grice (2006) looked at the accentuation of accessibility activated by different semantic relations (e.g synonymy, hypernym-hyponym, hyponym-hypernym, meronymy, etc...) and found that the disposition to deaccentuation (and the type of ToBI accent) depends on the semantic relation activating accessibility. So, for example, accessibility activated by antonymy (i.e., an antonym of the current word previously occurred in the discourse) or by part-whole tends to be more deaccented than that activated by whole-part or hypernym-hyponym.

We have seen that givenness (and accessibility) of an entity can be determined by a previous occurrence of an entity closely related (in a semantic sense) to it. Thus givenness could be seen as statically determined in the sense that the relation between the current entity and the previous entities always holds independently of the discourse context. However the givenness of an entity and its relation to closely related semantic entities can be "altered" by the discourse context. In the example below (taken from Calhoun (2006)):

(2.7)

S1:	Arun looked around all the fancy car shops - Mercedes, Porsche, BMW, Lamborghini ...
S2:	So what did he buy?
S3:	Arun bought a PORSCHE
S4:	What colour did he get?
S5:	Arun bought a RED Porsche
S6:	What did Joel buy?
S7a:	Joel bought a GREEN Porsche
S7b (wrong):	Joel bought a green PORSCHE
S7c:	Joel bought a green MERCEDES
S7d(wrong):	Joel bought a GREEN Mercedes

both "Porsche" and "Mercedes" have been mentioned in S1, so in the possible answers to S6 they are both given. However "Mercedes" has to be accented while "Porsche" can not (as shown in S7b). This difference in accentuation of the two words is due to the fact that "Porsche" is given in relation to an item of the discourse, i.e., the predicate "bought", which restricts the "Porsche" entity to the "bought Porsche" entity , which is given in S7, while there is not mention of "bought Mercedes" in S1-S6.

2.2.3 Beyond ±accents

The relation between Information Structure and prosodic prominence, and more in general between Information Structure and prosody, is not limited to pitch accent location and to the location of the main accent in an utterance. Information Structure concepts such as thematic and rhematic foci have been claimed to be signalled by different accent types. Contrast has been claimed to be marked by particularly prominent accents. Moreover Information Structure also affects prosodic structure.

The following section 2.2.3.1 concerns with the definition of the concepts of *focus/background*, *theme/rheme*, *contrast*. The relationship between these concepts and prosody is discussed in 2.2.3.2.

Concepts like *theme/rheme* are not directly relevant in this thesis as there is no attempt here to identify and use them in TTS synthesis but are mentioned to give a complete picture of the relation between prosodic prominence and information structure. On the other hand the concept of *contrast* is much more relevant for this thesis.

2.2.3.1 *Focus/Background*, *Contrast* and *Theme/Rheme*

In section 2.2.1 we have seen examples of *focus* but we have not given any definition of it. An attempt to give a definition of focus arises the question of whether there is one unique type of focus or multiple types. It has long been debated whether there is a unique ordinary focus, or whether there is also a contrastive focus separate from it. There are two main divergent accounts on focus: a semantic account and a syntactic account.

According to the semantic account, whose origins date back to Bolinger's work (Bolinger (1961)), there is no distinction between ordinary and contrastive focus because any focus is ultimately contrastive. In fact focus always evokes a set of alternatives and when this set "is narrowed down we get closer to what we think of contrastive accent"[8] Bolinger (1961). Bolinger's insight has been formalized by Rooth (Rooth (1992)) in his Alternative Semantics theory. According to Rooth the set of evoked alternatives must be constrained by context and its size can obviously range from large to small. However, a small (restricted) set does not require a semantic definition of a different type of focus and so there is no need for a categorical distinction between ordinary and contrastive focus.

The only dichotomy *focus/background* does not allow to capture all the concepts that determine the structure of information, like, for example, the *new/given* distinction, so that new dichotomies (often referred to as dimensions) have been proposed. Probably the most popular account on Information Structure's dimensions (and the most studied in term of prosodic correlates) is the one initially proposed by Halliday (1967) and subsequently refined by Vallduví and Vilkuna (1998) and Steedman (2000). They propose two dimensions of Information Structure consisting of *kontrast/background*, which is the equivalent of *focus/background*, and *theme/rheme*, which distinguishes between "the part of the utterance that relates it to the discourse purpose" and the part that "advances the discourse"(Kruijff-Korbayova and Steedman (2003)). *Kontrast* and *background* act within *theme* and *rheme* as shown in the following example from Kruijff-Korbayova and Steedman (2003):

(2.8) Q: I know that this car is a Porsche
But what is the make of your other car?

[8]Following this definition, in this thesis we use the term "contrastive focus" to refer to a focus that evokes a restricted set of alternatives. That does not imply we are assuming that contrastive focus deserves its own semantic category.

A :	My	other car	is	also	a Porsche
	background	kontrast	background	kontrast	background
	theme		rheme		

Thematic kontrast might be seen as contrastive focus (meant as focus evoking a restricted set of given alternatives). However it is easy to find examples where also *rhematic kontrast* is contrastive like in the following question-answer pair:

(2.9) Q: What did Paul and Mark buy?

A: Paul bought a Porsche while Mark bought a Maserati

Moving from the semantic to the syntactic account of focus, according to the syntactic account which dates back to Chomsky (1971), ordinary focus and contrastive focus are two different types of focus. Kiss (1998), for example, claims that contrastive and ordinary focus, in her terminology, exhaustive and informational focus respectively, have different semantic and syntactic properties and so they must be kept separated. An element focused by exhaustive focus is the only element picked from a set of alternatives which "achieves a true proposition when combined with the background" (Umbach (2004)). In English exhaustive focus may require word order movement as in it-cleft clauses. For example in "It was PAUL who did it", the exhaustive focus is on "Paul", since "Paul" and nobody else makes the proposition true (so, for instance, "It was PAUL who did it" can not be followed by "... and it was KARL who did it"). On the other hand, informational focus introduces new information, it does not involve movement and it does not evoke a set of given alternatives.

2.2.3.2 Prosodic Correlates of *Focus/Background*, *Contrast* and *Theme/Rheme*

The theme/rheme distinction has been claimed to strongly affect prosodic phrasing, with prosodic breaks aligned with theme/rheme boundaries (Steedman (2000)). So in the example above (2.8) the theme/rheme boundary occurring after "car" can be signalled with a prosodic break[9]. It is however difficult if not impossible, to identify theme/rheme boundaries looking at text only. In fact, again in example (2.8), the verb "is" could be removed from rheme and attached to theme without violating the definition of theme/rheme, but in that case a break after "is" would sound much less appropriate.

[9]Note that here we are only considering the case where the theme contains *kontrast*. When the theme is unambiguously established in the context and so all its items are *background* then the theme boundary is less prone to be marked by a prosodic break

Thematic and rhematic kontrasts have also been claimed (see Prevost and Steedman (1994) and Steedman (2000)) to be prosodically signalled by different accent types as shown below

(2.10) Q: I know that this car is a Porsche
But what is the make of your other car?

A :	My	other car	is	also	a Porsche
	background	kontrast	background	kontrast	background
		L+H*	LH%	H*	LL%
	theme		rheme		

with thematic kontrast signalled by a L+H* accent followed by a "fall-rise" tune (LH%), and rhematic kontrast marked by H* accent followed by "fall" tune (LL%).

Also contrastive focus has been claimed to be prosodically signalled with increased prominence or even emphasis, so that accents marking contrast has been referred to as contrastive accents to distinguish them from weaker "standard" accents. Obviously the prosodic correlates one associates to contrastive focus strongly depend on what one means with contrastive focus, and we have just seen that there is not a standard definition of contrastive focus. In most of the studies on the prosodic correlates of contrastive focus, contrastive focus is a "very restricted" focus where the set of alternatives is limited to two elements, that is two words (or phrases) that explicitly contrast with each other as in the examples below:

(2.11) S1: ... *So SAM had ravioli*
S2: No, KEN had ravioli.

(2.12) S: *John only paid for the BEER, not for the PORT*

As already stated in Chapter 1 we refer to *contrast* as the relation that links two items that explicitly contrast with (evoke) each other (as in example 2.12) or two items where one item contrasts with(evokes) the other (as in example 1.1). In order to avoid phrases like "focus activated by *contrast*", we will also use the term *contrast* to indicate the focus on the contrastive word (i.e., the word that evokes the other word).

Concerning the prosodic correlates of *contrast*, as we have mentioned above the accent on *contrast* (i.e., contrastive accent on contrastive words) is usually more prominent than the "standard" accent. It is however not clear how this increased prominence is realized, whether the distinction "standard" vs. contrastive accent is categorical (i.e., the two accents are phonologically separated with contrastive accent being represented

by L+H*), or whether it is not categorical and the perceived increased prominence is only due to the prosodic context (see Krahmer and Swerts (2001), for example).

Compared to other "types" of focus (and to other Information Structure's concepts), *contrast* seems to be easier to identify in that the alternatives evoked by the contrastive words are explicitly given.

Both phonetic correlates of *contrast* and its "explicit" nature make *contrast* a very interesting concept for TTS applications. For this reason a whole chapter of this thesis (chapter 5) is devoted to the automatic detection of *contrast* from text (actually to the detection of the "symmetric" *contrast*, i.e., *contrast* where the two words are both focused in that they contrast with each other).

2.3 Broad Focus and Pitch Accents

In sections 2.2.1 and 2.2.2 we have seen the main mechanisms through which discourse context affects the placement of pitch accents (and the placement of the primary accent). Discourse context is not the only factor affecting pitch accents distribution and the placement of the primary accent. Other factors affect accent placement, and are necessary when the influence of discourse context on pitch accenting is weak, as in the presence of broad focus[10]. Some of the factors we will review have been claimed to mainly determine the placement of the primary accents, some others to affect word accentability (without a distinction between primary and secondary accent). The relation of some of these factors to pitch accenting have been claimed on the basis of some "linguistic insights" supported by weak experimental evidence (i.e., observations from carefully selected utterances), some others are supported by stronger experimental evidence (i.e., statistical analysis from prosodically annotated speech corpora).

Most of the discourse-independent factors affecting pitch accent placement account for the degree of informativeness of words where the degree of informativeness (i.e., "the amount of unpredictable information") is intrinsic (defined regardless of any context) or dependent on sentence-internal words. For example, high-frequency words, which usually convey a large amount of predictable information, are much less prone to accentuation than low-frequency words.

Similarly, *semantically empty* content words, such as *person*, *man*, *thing*, *place*,

[10] As Ladd points out in Ladd (1996) the accent pattern within a broad focus is not "contextless', but only 'unmarked', that is the pattern that is chosen when there is no compelling grammatical or contextual reason to choose some other'.

and so on, are often not accented. Semantic weight is an intrinsic property of some nouns that, according to Bolinger (1972) can be illustrated comparing empty nouns with rich nouns as in the following example (with capitalized words indicating the last nuclear accent in the sentence):

(2.13) S1: He was arrested because he KILLED a man

(2.14) S2 : He was arrested because he killed a POLICEMAN

The degree of informativeness of a word can also be relative, i.e., affected by the words surrounding it. As Bolinger (1972) notes, in the following example:

(2.15) S1: They STRANGLED him to death

(2.16) S2 : They hounded him to DEATH

in S1 "death" is deaccented because "strangulation" implies "death" and so makes "death" more predictable and consequently less informative, i.e., "death" adds very little information to the sentence. In S2 "hounded" does not necessarily imply death so "death" is more informative (than in S1) and is not deaccented (and so it bears the nuclear accent).

Pan and McKeown (1999) and Pan and Hirschberg (2000) propose an information-theoretic measure of intrinsic informativeness and relative informativeness respectively. The intrinsic informativeness of a word is defined in Pan and McKeown (1999) as:

$$IC(w) = -\log(p(w)) \tag{2.1}$$

where $p(w)$ is the probability of a word w of appearing in a corpus. The lower the probability of a word, the higher its Information Content (IC).

Note that, as we will point out in chapter 3, this information-theoretic definition of informativeness does not entirely account for the concept of semantic weight (in fact a high IC word can be semantically empty). For such reason in chapter 3 we propose an information-theoretic definition of semantic-weight that is complementary to IC in the measuring of intrinsic informativeness.

A measure of relative informativeness is proposed in Pan and Hirschberg (2000), where the relative informativeness of a word is expressed as :

$$RIC(w_i) = -\log(p(w_i|w_{i-1})) \tag{2.2}$$

where $p(w_i|w_{i-1})$ is the probability of observing a word given the occurrence of the previous word (RIC stands for Relative Information Content). This is obviously a

a. General schemata:

Furniture + Room → RIGHT

(e.g., *kitchen* ***table***)

b. Schemata with particular head nouns:

Proper-name + *Street* → LEFT

(e.g., ***Park*** *Street*)

c. List of particular Complex Noun Phrases:

White + *House* → LEFT

Figure 2.3: *Sproat's types of accent placement rule, from Sproat (1994)*

measure of a simplified version of relative informativeness that, for example, does not apply in cases like example (2.15).

Both IC and RIC have been shown to have a positive correlation with pitch accent assignment to a word and to be useful features for pitch accent prediction.

Nevertheless there are cases where the relation between informativeness and accentuation does not hold. For example informativeness does not explain why while in the noun phrase "apple cake" the accent is on "apple", in the noun phrase "apple pie" the accent is on "pie".

Sproat (1994) proposes a set of rules to assign pitch accents in two-word noun phrases to handle cases where accenting is determined by lexical effects. The rules can be roughly divided into three different types as shown in figure 2.3.

Also syntax has been claimed to affect accenting. Predicates have been claimed to be less accentable than their arguments. For example in short sentences describing single events, the accent (or the main accent) tends to be on the subject as it is shown in the examples below (from Ladd (1996)):

(2.17) My UMBRELLA broke

(2.18) S1: The SUN came out

(2.19) S1: His MOTHER died

By contrast, "if the subject denotes a human and the predicate denotes an action over which the subject is likely to have some control the accent on the verb is more likely" (Ladd (1996)):

(2.20) My brothers are WRESTLING

(2.21) S1: Jesus WEPT

(2.22) S1: The professor SWORE

Finally, as we have already briefly mentioned above, rhythm affects pitch accent placement by forcing the occurrence of pitch accents on long phrases that otherwise could be entirely deaccented. Rhythm also indirectly may affect the location of nuclear accents by requiring utterances to be split into prosodic phrases (delimited by prosodic breaks), that in turn have to have their own nuclear accent. Obviously rhythm is not the only factor affecting prosodic phrasing, also other factors, such as Information Structure and syntax affect it, and so through prosodic phrasing they affect, again, the placement of nuclear accents.

Experimental evidence from previous work on automatic pitch accent prediction (see section 2.4) shows that all the factors affecting accenting we have just seen do not affect accenting deterministically. There are not deterministic IF-THEN rules determining where accents have to be placed. In fact so far we have often used the expression "words with this property *tend* to be deaccented/accented". As we will see in much more detail in Chapter 4 uncertainty plays a considerable role in the placement of accents and it is what allows a sentence to be uttered using different (but equally acceptable and conveying the same meaning) accentuation patterns.

Such uncertainty is mainly due to effects that go beyond what can be inferred by only looking at text. Part of such uncertainty is "explained" by the fact that each human speaker has her own free will and that her speech is subject to peculiar physical constraints of her production system.

Another part of such uncertainty is due to disagreement among transcribers on the location of pitch accents. Studies on inter-transcriber agreement on accent location using ToBI conventions report reasonably high agreement rates ranging from 80% to 90%. However as Ladd points out in Ladd (2009) there are types of situations where disagreement is not uncommon. In those cases the ToBI conventions seem to be underspecified (and the annotation examples provided to the transcribers seem contradictory) so that the annotation mainly relies on the intuitive judgments of the transcribers that can be biased by different types of cues (e.g., acoustic vs. metrical, i.e., based on the prominence tree mentioned in section 2.1). Being left to subjective judgment, the annotation process becomes prone to variability, which in turn leads to uncertainty.

Additionally, even if all transcribers used the same type of cues, variability could still occur since the annotation of prosodic symbols is a process of abstraction and dis-

cretization of prosody that relies on the perceptual apparatus of the human annotators which is not identical for all annotators.

Since prosodic prominence is not just about ± accent we believe that all factors that have been claimed to affect accenting, actually do not directly affect accenting, but affect the relation of relative prominence among words in an utterance. That is another reason that partially explains why there are not deterministic rules in pitch accent placement.

If we accept the notion of prominence tree discussed in section 2.1, then we can accept the fact that some factors, e.g., focus, rather than affecting accenting actually affect the weak-strong relations in the prominence tree making some syllables stronger than they would be in a "default tree". The resulting strong syllables then can be accented or not and some accents can be stronger than others. The weakest accents can be perceived or not.

On the other hand, some factors, mainly intrinsic properties of words like intrinsic informativeness, seem to determine the accentability of a word regardless of the weak-strong relations of that word with the other words. These are the factors that are mainly taken into account in automatic pitch accent prediction.

2.4 Automatic Pitch Accents Prediction and Detection

Most fully automatic pitch accent predictors (i.e., predictors in which both accent prediction and feature extraction are automatically carried out) relies on training features that are easily computable and are not related to discourse context. The most used and effective features are Part-of-Speech, Information Content, and Relative Information Content. Yuan et al. (2005) show the effectiveness of what they call the *accent-ratio* feature, a feature that gives a measure of how a word is prone to be accented simply by counting (and using a bit of smoothing in Brenier et al. (2006)) how many times that word has been accented in a prosodically labelled corpus. Other non-discourse-related but less effective features are "phonological" features such as (from Gregory and Altun (2004)): number of canonical features in the word, number of canonical/transcribed phones in the word, length of the utterance containing the word, position of a word within the utterance.

The studies mentioned above are concerned with the prediction/detection of accented words, but there are also studies concerned with the prediction/detection of accented syllables (Ross and Ostendorf (1996), Sun (2002), and Levow (2008) for

example).

Here we refer to accent prediction when accents are predicted from text (and so only textual features can be used), while we refer to accent detection when accents are identified in speech (and both acoustic and textual features can be used). In accent detection, textual features usually turn out to be more predictive than acoustic features (in Chen and Hasegawa-Johnson (2004) and Levow (2008), for example) although it is difficult to make a fair comparison since that depends on the set of textual and acoustic features one uses. Usually the combined use of acoustic and textual features leads to a very small increase in detection accuracy with respect to detection based on textual features only (see Levow (2008) and Sridhar and Bangalore (2008), for example)

Pan et al. (2002), which probably is the study using the most complete set of features, show that semantic (e.g., semantic role) and rich syntactic (e.g., info extracted from syntactic constituents, e.g., noun phrases, verb phrases, etc...) information provided by a Natural Language Generator are significantly correlated to pitch accent placement but they are useless when used in combination with the whole set of features to predict pitch accent placement. IC and RIC are the most predictive features while surprisingly POS is superfluous. The information status of a concept (and of the words conveying that concept), is, after IC and RIC, the most predictive feature.

On the other hand Sridhar and Bangalore (2008) show that the use of syntactic supertags, i.e., word-level syntactic tags embodying predicate-argument information, improves accuracy in pitch accent prediction.

To the best of our knowledge Hirschberg (1993) is the most serious attempt and the only entirely automatic work taking into account context to predict accent placements. Making use of Grosz and Sidner (1996)'s model of discourse which divides the discourse structures into *Linguistic, Attentional* and *Intonational* structure, Hirschberg models the Attentional structure (whose definition is very close to the definition of Information Structure used in this thesis) as the union of a *global focus space* and a *local focus space*[11]. *Global focus space*, which should contain central concepts of the discourse which are relevant throughout the whole discourse, is modeled as a static set of all content words in the first sentence of the text (i.e., discourse), while *local focus space*, which contains concepts that are currently relevant in the discourse, is modeled as a stack of words that contains the roots of words that are currently processed and it is popped in presence of *cue phrases* (e.g., now, well, by the way) or paragraph

[11]Note that the term focus in Grosz and Sidner (1996) and the term we have used so far and use throughout this thesis have two different meanings.

boundaries. If the root of the current content word is already present in the *local focus* then the word is given, otherwise is new. If a new word is in the *global focus* then is tagged as contrastive. Hirschberg shows that the contribution of these discourse context related features to pitch accent prediction on BURNC is, on the author's account, disappointingly smaller than expected.

Brenier et al. (2006) show that the contribution of hand-labelled information status and focus in spontaneous spoken dialogues (in section 2.5 we will see details of the annotation of focus used by Brenier et al. (2006)) is small compared to features like *accent ratio*, IC and POS.

This result certainly "resizes" the actual impact of Information Structure on pitch accenting and, at a first glance, it might seem to contradict some of the claims about the relationship between Information Structure and prosodic prominence. However a couple of considerations have to be made. First, as we have already discussed, *narrow focus* has been claimed to determine the placement of the primary accent in a prosodic phrase rather than determining word accentability. Second, in sections 2.2.3.1 and 2.2.3.2 we have seen examples of carefully designed sentences where the distinction focus/background and new/given can be easily carried out, while when utterances are not preceded by questions (which is often the case in spontaneous spoken dialogues) these distinctions are much harder to identify in that the answer (i.e., focus) within each utterance is no longer triggered by an explicit question (but by an implicit question that summarizes the discourse history). Thus since Information Structure becomes more "blurred" we expect its influence on prosodic prominence to become weaker. Perhaps in more "controlled" dialogues (e.g., Wizard-of-Oz dialogues, where a human user believes is talking to dialogue system that is actually a human) where most of the utterances are questions or answers to those questions, and answers are generally not over-informative, the benefits of taking into account Information Structure to predict pitch accent placement are more substantial.

Interestingly none of the cited studies, apart from Ross and Ostendorf (1996), Calhoun (2006) and Calhoun (2008), goes beyond the prediction of $\pm$accent and/or ToBI accent type. However we have seen in section 2.2.1 that a rough $\pm$accent distinction does not entirely account for the relation of prosodic prominence between words.

Ross and Ostendorf (1996) reported that prediction of more than two categorical levels of accentual prominence (i.e., $\pm$accent) is not feasible since human annotators can not consistently annotate more than two categorical levels of accentual prominence. So they tried to predict prominence as "a continuous-valued normalized F0

peak for each accented syllable" by using linguistic features. The ToBI accent type turned out to be the most predictive feature followed by other phonological features.

Calhoun (Calhoun (2006) and Calhoun (2008)) shows a good agreement between annotators in the labeling of nuclear accents and shows that focus and the location of prosodic breaks are the best predictive features to distinguish between nuclear and non-nuclear accents while they are of little utility in the $\pm$accent detection task.

In chapter 3, where $\pm$accent prediction is addressed, we propose new predictive features that are complementary to the best features proposed in the literature in the representation of the intrinsic and sentence-internal factors affecting pitch accenting. The focus on these features rather than on discourse-related features is justified by the primacy of intrinsic and sentence-internal factors over discourse-related factors in the influence on pitch accenting.

Finally, previous work on pitch accent prediction/detection has not only focused on the extraction of the best features to improve prediction/detection accuracy but has also addressed the issue on the most suitable machine learning techniques for accent prediction/detection.

Classification trees are by far the most used ones. Sun (2002) uses two ensemble machine learning methods, bagging and boosting, and Classification trees as basic learning algorithm within the two methods. Classification trees within ensemble methods outperform the Classification tree alone.

The use of classifiers such as Classification Trees, Neural Networks, Logistic Regression and so on, implies the assumption that the probability of the placement of a pitch accent is independent of the placement of the surrounding pitch accents. This independence assumption could actually result in an oversimplification of this classification task. In fact if we suppose to have a prosodic phrase containing more than 4/5 words all of them having a high probability of being accented, it is quite unlikely that all of them will be accented.

Classifiers based on Hidden Markov Models (HMM) (Baum and Petrie (1966))and Conditional Random Fields (CRF) (Laffeerty et al. (2001)) allow to relax such independence hypothesis. Pan and McKeown (1999) show that first-order HMM classifier achieves better results than RIPPER (Cohen (1995)), a rules-learner classifier whose performance is comparable with that of Classification Trees. Gregory and Altun (2004) show that CRF outperform HMM on the Switchboard corpus.

However previous work lacks of a in-depth investigation of the impact of the dependence/independence assumptions which is one of the main issues addressed in chapter

3, the chapter devoted to ± accent prediction where accent predictors based on Classification Trees, ensemble methods, HMM and CRF are compared.

2.5 Automatic Detection of *Contrast* and Other Information Structure Concepts

The main obstacle one has to face when trying to work on the the automatic detection of Information Structure concepts is the very limited number of annotated corpora available. Moreover all these few corpora differ in terms of language (e.g., English vs. German), dimensions of Information Structure adopted, annotation conventions, etc...

Postolache et al. (2005) automatically detect Topic and Focus (which respectively corresponds to the theme and rheme defined in section 2.2.3.1) from the Prague Dependency Treebank (PDT), which consists of Czech newspaper articles. They use manually labeled training features from PDT which consists of 1) syntactic functions, like subject, object, predicate, and so on; 2) attribute of the nodes of tectogrammatic trees, where tectogrammatic trees are comparable to syntactic dependency trees (see section 3.2.2) that only contain "autosemantic" words, i.e., content words, and node attributes are semantic roles such as actor, patient, addressee. In addition they also use features derived from the Topic/Focus annotation guidelines. The accuracy achieved using a Classification Tree is very high, 90.7% .

Zhang et al. (2006) automatically label what they call *kernel focus* and *symmetric contrast* by training their labeler on a Wizard-of-Oz corpus collected in a tutoring dialogue scenario. This is an "ideal" corpus for focus detection since it mainly consists of question-answer pairs in a limited-domain scenario. Their *kernel focus* is very similar to narrow focus (both thematic and rhematic) while *symmetric contrast* corresponds to a subtype of our *contrast*, that is a *contrast* activated by syntactic parallelism. They achieve high accuracy by using a combination of acoustic features (F0, duration, energy and spectral balance cepstral coefficients), Part-Of-Speech, and a semantic similarity measure computed by using the WordNet semantic lexicon and corpora statistics. The very high accuracy, especially in the symmetric-contrast detection task, is largely explained by the very small size of the dialogues domain.

In Nenkova et al. (2007) a subsection of the Switchboard corpus annotated by Calhoun et al. (2005) is used to detect all the annotated "categories" of focus (referred to as *kontrast* in Calhoun et al. (2005)). Word and phrases were marked as focused

if they sounded salient to the annotators (that could listen to the utterances). Focus categories are not meant as actual categories but scenarios in which focus can occur. These categories are:

- *correction*. the focused word/noun-phrase corrects a previous word/noun-phrase.
- *contrastive*. The focused word/noun-phrase explicitly contrast with another word/noun-phrase.
- *subset*. "The [focused] word/noun-phrase is (a) a current topic, and (b) a member of a more general set mentioned in the context"Calhoun (2006).
- *adverbial* "The speaker used a focus-sensitive adverb (i.e., "only", "even", "always", "especially", "just", "also" and "too") to highlight the focused word/noun-phrase and no another words/noun-phrases from a plausible set (which does not need to be explicit)" Calhoun (2006)
- *answer* "The [focused] word/noun-phrase is an answer if it, and no other, filled an open proposition set up in the context by either speaker"Calhoun (2006).
- *other* the word/noun-phrase is clearly focused but the type of focus does not fall in any of the previous categories

Note that focus items may belong to more than a category. The union of *correction* and *contrastive* corresponds to what so far we have referred to as *contrast*.

The focus tagger proposed by Nenkova et al. (2007) distinguishes between *kontrast* and *background*, where *kontrast* gathers all the focus categories mentioned above (e.g., adverbial, answer, etc...). It looks at the acoustic properties, POS and *accent ratio*. The tagger outperforms a majority class (background) baseline. This surprising result (given the simple syntactic feature used and the absence of semantic features) is due to the fact that some POS, i.e., nouns and adjectives, are mainly *kontrastive*. No attempt to distinguish the different categories is made.

Using the same corpus, Sridhar et al. (2008) attempt to detect each single category of focus (vs. background). They use the same features proposed in Nenkova et al. (2007). They report a 71.10% accuracy in classification accuracy between background and the *contrastive* category achieved by only using POS. Again, this result is due to the fact that some POS are mainly *contrastive*. However such an approach to *contrast* detection does not seem useful, at least for TTS purposes, since it does not guarantee an acceptable precision on the POS that are mainly *contrastive*, and it cannot detect

the *contrast* relation (i.e., which word contrasts with which word?), which we believe is essential for a correct and consistent prosodic realization of *contrast* in speech.

2.6 Text-to-Speech Synthesis

A Text-to-Speech (TTS) system, a system that converts raw text into speech, consists of two main components: a Text Analysis module, and a Speech Synthesis module.

The Text Analysis module first normalizes the raw text by expanding acronyms, numbers, abbreviations into "well-formed" words (e.g., "11" would be expanded in "eleven"), and subsequently converts words into "streams" of linguistic information that will then be input to the Speech Synthesis module. The most important stream is the phonemic stream containing the sequence of phonemes generated by a Grapheme-to-Phoneme (G2P) function [12]. Other common linguistic information streams are streams indicating the presence of stress (lexical stress), the position of phonemes in their respective syllables, words, intonational phrases, distances of phonemes to syllable boundaries, distances of syllables to word boundaries and so on.

The Speech Synthesis module uses the linguistic information to generate appropriate speech. There are different techniques to synthesise speech, most of them can be roughly grouped into two approaches. One approach can be regarded as a memory-based approach (as suggested by Taylor (2009)) in which speech is generated by selecting and joining units of recorded natural speech. Signal processing techniques may be applied to smooth discontinuities at the joint points.

In the alternative approach (referred to as "learning-based" approach in Taylor (2009) as opposed to the memory-based approach[13]) recorded natural speech is used to extract acoustic properties of speech (that could be formants, or Mel cepstral coefficients, or something else depending on the type of representation of speech one chooses) and then to learn (manually or by machine learning techniques) their behaviour with respect to the linguistic information. During synthesis, the reverse process is applied so the predicted acoustic properties are converted into speech by mean of synthesis filters.

In sections 2.6.1 and 2.6.2 we will briefly describe the two most successful realiza-

[12] Although the G2P process could theoretically by-passed and graphemes could be directly used as a stream of linguistic information

[13] A more accurate name for this approach could be "acoustic properties learning-based approach", since also in articulatory synthesis, a technique that does not fall in neither of the two approaches mentioned here, a mapping between articulatory features and speech is learnt.

tions of the "memory-based" and the "learning-based" approach respectively.

Most of the people working on TTS are mainly concerned with the improvement of the Speech Synthesis module, while comparably little research work has been done on the Text Analysis side and it mainly concerns G2P issues. Very little work has been done to find out the impact of prosody related linguistic information (ranging from pitch accents to syntactic categories) on TTS quality. One of the aims of this thesis is that of investigating the utility for TTS synthesis of linguistic information, mainly syntactic and semantic information, related to prosodic prominence.

2.6.1 Unit Selection Speech Synthesis

The unit-selection technique is the last generation technique of the "memory-based" approach. It differs from previous generation "memory-based" techniques, such as the so called diphone synthesis, in that the speech database is much larger and usually contains several units (e.g., diphones) having the same phonemic specification (e.g., several occurrences of the diphone /k-a/). This abundance of "replicas" allows to greatly reduce the need of signal processing which usually heavily degrades the naturalness of synthetic speech.

Depending on the speech synthesis system used, the basic unit type, i.e., the shortest unit type available, can be half-phone, phone, diphone, syllable, etc...

The critical issue in unit selection is finding out the best way to select the best sequence of speech units.

Almost all unit-selection systems are based on the algorithm proposed by Hunt and Black (1996). In this algorithm each speech unit (diphones in Hunt & Black, but that is not a constraint) of the speech database is represented (and so identifiable) as a vector of linguistic features (u_t). During synthesis, the sentence to synthesise is transformed in a sequence of vectors of linguistic features (s_t). These vectors contain exactly the same features contained in the vectors representing the speech units so that the sentence is "uttered" by selecting speech units that have same (or similar) vectors as those specified for the sentence (target vectors). A cost, called target cost ($T(s_t, u_t)$), is computed to quantify how close the target vector and the speech unit vector (candidate vector) are. An additional constraint is that adjacent selected speech units should join well, that is they should not generate perceivable distortion when joint together to generate the utterance. As a consequence another cost, called joint cost ($J(\tilde{u}_{t+1}, \tilde{u}_t)$), is computed to measure how well two speech units join. Note that u_t and $\tilde{u}_t$ refer to the

same speech unit but to different feature vectors because the features used to compute the joint cost are different from those used to compute the target cost.

The target cost function $T(s_t, u_t)$ is usually defined as a weighted sum of distances between the two units with respect to the linguistic specifications (i.e., the values of the linguistic features):

$$T(s_t, u_t) = \sum_{f=1}^{F} w_f(T_f(s_t[f], u_t[f])) \tag{2.3}$$

where $s_t[f]$ and $u_t[f]$ are the values for the feature f of the target unit and the speech unit respectively, T_f is the function evaluating the distance between $s_t[f]$ and $u_t[f]$, and w_f is the weight of the feature f. Usually T_f is defined as follows:

$$T_f = \begin{cases} 0 & \textit{if } s_t[f] \textit{ and } u_t[f] \textit{ are equal} \\ 1 & \textit{otherwise} \end{cases} \tag{2.4}$$

The weights account for the fact that some features are more important than others in the evaluation of the perceptual similarity of two speech units.

The definition of $T(s_t, u_t)$ given in section 2.3 is a "classic" definition that assumes the contribution of each feature is independent of the others. An alternative formulation, the acoustic space formulation, in which no independence assumption is made, can be alternatively used.

The total cost (i.e., target + joint cost) of a sequence of candidate speech units is defined as:

$$C(S,U) = \lambda \sum_{t=1}^{t=OT} T(s_t, u_t) + (1-\lambda) \sum_{t=1}^{t=OT-1} J(\tilde{u}_{t+1}, \tilde{u}_t) \tag{2.5}$$

where OT is the length, in number of units, of the utterance and λ is used to give a different weight to the target and the joint cost.

The goal of the unit selection algorithm is finding the sequence of speech units $\hat{U}$ that minimizes $C(S,U)$ that is:

$$\hat{U} = \arg\min_{U} C(S,U) \tag{2.6}$$

This is usually achieved by using the Viterbi algorithm integrated with a pruning technique (usually the beam pruning) to reduce the, otherwise huge, search space.

An important issue in unit selection systems is the creation of an appropriate speech database, which has to comply with the constraint of being phonetically rich, i.e., it has to contains all the phones of the phonological system of the language of the speaker

and at least the most frequent sub-sequences of those phones (e.g., the most frequent sequences of two phones).

While the constraint of having a phonetically rich database is feasible, the requirement on the database of also being prosodically rich would lead to an enormous and so intractable database. Unfortunately the use of signal processing techniques to make up for the lack of prosodic richness of the database does not produce the desired effect. For that reason, perhaps the main limit of unit-selection speech synthesis is the inability to scale up to prosodic styles other than the "neutral" style in read speech.

2.6.2 HMM-based Speech Synthesis

Instead of storing several realizations of a unit (e.g., hundreds of realizations of the phoneme /a/) in the speech database we can use a model of it expressed as a probability density distribution of some acoustic properties (usually referred to as acoustic coefficients), for example a mixture of multivariate Gaussians with 13 mel-cepstral coefficients as variables.

However a mixture of multivariate Gaussians is usually not enough to model all the variability in the realisation of a phoneme, so more complex models are necessary.

In HMM-based speech synthesis (Tokuda et al. (2000)) phones are modelled using phone HMM-models as the one shown in figure 2.4. A HMM phone model usually consists of three or five hidden states to account for the fact that the values of the acoustic coefficients at the beginning and end of a phone are quite different from those in the middle of a phone, due to co-articulation effects. An output probability density distribution (usually a mixture of multivariate Gaussians) of the vector of acoustic coefficients $\mathbf{o}$ (observation vector/state) is associated to each state. The probability of moving from one state to another depends on the transition and the self-transition probabilities.

A single HMM-model for a phoneme does not account for the fact that realizations of the same phoneme can be very different because of co-articulation effects and different linguistic properties (e.g., one phone is stressed and the other is unstressed). However having a HMM model for each phonemic context plus linguistic properties is unfeasible because of data sparsity problems (i.e., there would be too few or zero examples for each model). To solve this problem tree-clustering is applied in order to cluster speech units (e.g., the phone onsets if we use a three-state phoneme model) in the acoustic space. The clustering, usually referred to as context-sensitive clustering,

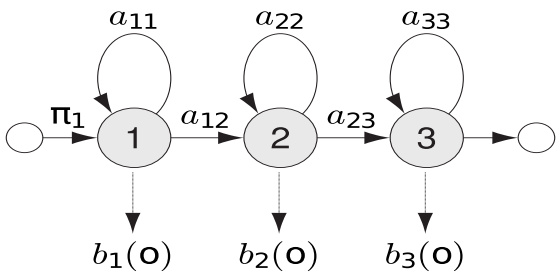

Figure 2.4: *A 3-state HMM phone model.* a_{12}, a_{23},... *are transition probabilities.* a_{11}, a_{22},... *are self-transition probabilities and* $b_1(o)$, $b_2(o)$ *are density probability distributions of the acoustic coefficients.*

is carried out searching for the "questions" on the linguistic properties (e.g., "is the previous phone a nasal?") that mostly determine the acoustic similarities/differences among speech units. As a consequence a system using three-state models will have three trees, whose leaves will have their own probability density distribution of the acoustic coefficients.

In a HMM-based speech synthesis system the acoustic space is represented both by static acoustic coefficients (e.g., mel-cepstral coefficients, logF0,...) and dynamic coefficients that indicates how the static coefficients evolve over the time. The use of dynamic coefficients is necessary to avoid unnatural discontinuities when generating the values of the acoustic coefficients during synthesis.

Once the HMM-models have been trained they can be used for speech synthesis. The main task consists in finding the optimal sequence of vectors of acoustic coefficients $\mathbf{O} = (o_1, o_2, ..., o_T)$. The optimal sequence can be defined as:

$$\mathbf{O}^* = \arg\max_{\mathbf{O}} \sum_{all\,\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda, T) \tag{2.7}$$

where $\mathbf{q}$ is a sequence of states, λ is the set of the parameters of the HMM-models (it contains state-transition probabilities and output probability distributions), and T is the duration (in number of frames) of the utterance to be synthesised.

Since there is not known analytical close solution for equation 2.7, an approximated

solution is computed by first computing the best sequence state $\mathbf{q}$, that is the sequence:

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} P(\mathbf{q}|\lambda, T) \tag{2.8}$$

and then computing the best sequence of vectors of acoustic coefficients:

$$\mathbf{O}^* = \arg\max_{\mathbf{O}} P(\mathbf{O}|\mathbf{q}^*, \lambda, T) \tag{2.9}$$

The resulting $\mathbf{O}^*$ is then passed to the synthesis filters (source and vocal tract filters) to generate speech.

2.7 Modelling Prosody in (TEXT-to-) speech synthesis

The modelling of prosody in unit-selection speech synthesis can be either implicit or explicit while in HMM-based speech synthesis is exclusively explicit.

In several unit-selection systems there is not explicit modelling of the acoustic correlates of prosody (i.e., F0, duration and energy), so prosody is implicitly modelled by adding in the target cost function linguistic features (e.g., pitch accents) that are correlated with duration, intonation and energy. The cost weights to these features determine the weight prosody has in the selection of speech units. Alternatively F0, duration and energy values can be predicted from the linguistic information and then used as target cost specifications (in place of the linguistic information from which they have been predicted).

In HMM-based speech synthesis state durations may be implicitly modelled by state transition and self-transition probabilities. However this solution would lead to unrealistic models of duration and so state durations are explicitly modelled by Gaussians (and clustering is applied as for spectral features).

To model F0, HMM-based speech synthesis uses the concept of streams. The observation vector is divided into streams to account for the fact that spectral features and F0 have to be treated separately (and so have to belong to different streams) since F0 behaviour is described by density distributions that drastically change depending on whether the state is in a voiced region or an unvoiced region. The use of streams separating spectral features from F0 implies that separated context-sensitive clustering is applied for F0 only (dynamic F0 coefficients included).

Most of the (relatively little) research on prosodic modelling for speech synthesis is concerned with finding out successful methods to improve the prosodic quality of

synthetic speech given a set of linguistic features that are related to prosody, like POS, positional features (e.g., syllable position in the phrase), prosodic break placements, etc...

Very few studies are concerned with the use of new linguistic features and the analysis of their impact on the realisation of prosody in synthetic speech.

Pitrelli and Eide (2003) define the intonation contour of *contrast* in terms of ToBI labels, and then use these ToBI labels to predict F0 and duration (which in turn are used as features in the target cost function of the IBM unit-selection system) to generate manually identified *contrast*. Their approach slightly outperforms a default intonation baseline (where *contrast* is not prosodically signalled with specific accents).

Baker et al. (2004) directly use ToBI labels as target cost features to model thematic and rhematic focus and theme and rheme boundaries according to Steedman's theory (Steedman (2000)) in a limited-domain unit-selection speech synthesis system. Their approach outperforms a default intonation baseline only for some configurations of theme-rheme (for instance, theme followed by two rhemes).

Strom et al. (2007) use automatically predicted $\pm$accent and manually annotated "emphatic" accents as features in the target cost function. They report that the combined use of $\pm$accent and emphatic accents outperforms a default speech synthesis system that does not use them as features in the target cost function.

To the best of our knowledge no work, using either unit-selection or HMM based speech synthesis, is concerned with the automatic identification and use of linguistic features (such as *contrast*) that account for the effects of discourse/sentence context on the generation of prosody.

Chapter 3

Automatic Pitch Accent Prediction

This chapter has two main goals. One is that of pushing a bit further the accuracy of state-of-the-art pitch accent prediction by using new predictive features that cover effects on pitch accenting not covered (or only partially covered) by features proposed in previous work. The second goal is that of investigating which machine learning techniques are most suitable for accent prediction, in particular whether techniques used for sequential data (e.g., HMM-based or CRF-based predictors) are more suitable than techniques implying independence between the placement of one accent and the previous accent placements.

3.1 Predictive Features from Previous Work

From previous work on pitch accent prediction and detection it is hard to say what training features are most predictive. Almost all proposed accent predictor/detectors use different sets of features, so features that turned out to be predictive in one study may not be good features when combined with other features since the useful information they contain may already be encapsulated in other more predictive features. The comparison of features is also complicated by the fact that there are also differences in the training data used, features that are useful on one corpus might not be equally predictive on another corpus.

In spite of these difficulties it is still possible to draw some useful conclusions especially by looking at work that use sets of features accounting for different effects on accent placing (e.g., syntactic effects, discourse-context effects, etc...). For example, in Pan et al. (2002) and Brenier et al. (2006), it emerges that features that account for the intrinsic informativeness of words (e.g., Information Content) are among the best training features for pitch accent prediction.

These findings do not necessarily imply that other effects on accent placing are very minor compared to effects due to intrinsic properties of words but might imply that the modelling of other effects (e.g., discourse-context effects) is too gross.

3.2 New Proposed Features

In chapter 2 we saw all the main effects on pitch accenting. Here we just recall and categorize them.

- Discourse-context effects (see section 2.2)

  - focus/background dichotomy
  - given/new dichotomy

- Sentence-level effects (see section 2.3)
  - relative informativeness (e.g., probability of seeing a word given the words preceding it)
  - syntactic effects (e.g., predicate vs. arguments)

- "Intrinsic informativeness" effects (see section 2.3)
  - word informativeness

- Lexical effects (i.e., words may be accented or not depending on whether they are in compounds or not) (see section 2.3)

- "Phonological" effects, that is effects due to length of a word, its phonemic constituents, accent ratio, etc... (see section 2.4)

Note that this classification is not rigid in that some effects can be associated to more than a category. For instance the dichotomy content/function word (with content words much more prone to accentuation than function words) can be both considered as a syntactic effect and as an "intrinsic informativeness" effect.

In our search for new and predictive features we try to cover all the main effects on accent placing with a special focus on sentence-level effects and effects related to intrinsic properties of words as they have turned out to be the main effects on pitch accenting (see Brenier et al. (2006) for example, and discussion at the end of section 2.3).

The full set of training features used in this work consists of a set of features already proposed in the literature plus a set of new features. Most of these new features are new in the sense that they have never been used in accent prediction but have been already used in tasks very different from accent prediction.

The "old" features are:

- Information Content. $IC(w_i) = -\log(p(w_i))$ (where w_i is the word in position i and $p(w_i)$ is its probability computed off-line in a corpus)

- Relative Information Content. $RIC(w_i) = -\log(p(w_i|w_{i-1}))$

- Inverse Relative Information Content. $IRIC(w_i) = -\log(p(w_i|w_{i+1}))$

- Part-Of-Speech (POS)
- Word's Length (WL) (in number of characters)

IC, RIC and IRIC where computed on a corpus of 9 million words (from The Herald news) and using the CMU Language Model toolkit (Clarkson and Rosenfeld (1997)). We also used a bigger corpus (17 million words from The Herald news) but that did not improve the accuracy of the accent predictor (actually the accuracy slightly worsened). POS were obtained using the MXPOST tagger (Ratnaparkhi (1996)).

The new features are described and motivated in the following sections.

3.2.1 Information Content of Concepts (ICC)

To account for the degree of informativeness of words, a simple distinction between function vs. content words can be used. Function words like determiners, conjunctions and so on are essential for the syntactic well-formedness of a sentence but do not convey as much new (unexpected) information as content words do.

POS partially serve for the same purpose of distinguishing informative from non-informative words.

The information-theoretic definition of informativeness proposed by Pan and McKeown (1999), called Information Content ($IC(w) = -\log(p(w))$), is the most predictive feature among those accounting for word informativeness (see Pan et al. (2002), Brenier et al. (2006)).

However this definition of informativeness, by assuming that the informativeness of a word is only determined by the frequency of that word in a large corpus, fails to completely encode the degree of generality (or alternatively, specificity) of concepts associated to words, which is another factor related to our intuitive idea of informativeness. The more a concept is generic (or semantically empty, see section 2.3) , the more probable it is to occur (since it also occurs when concepts subsumed by it occur) and so the less informative it is. This is not necessarily encoded in IC. The words "cat", "feline" and "animal" might have similar frequency counts on a large corpus (for example, in the newspaper corpus used to compute the language model, the frequency of "cat" and "animal" are very close whereas "feline" never occurs in the corpus) but their degree of specificity is not similar. The concept of animal (expressed by the word "animal") is the most generic as it subsumes the concept of feline which in turn subsumes the concept of cat.

To measure the degree of specificity of a concept, we have to apply the information-theoretic definition of informativeness to the concepts conveyed by the words, rather than to the frequencies of the word tokens. In other words we need a measure that is related to the probability of seeing a concept rather than to the probability of seeing a word. Such a measure was proposed by Resnick (1995)[1]:

$$ICC(c_i) = -\log(p(c_i)) = -\log(\frac{freq(c_i)}{N}) = -log(\frac{\sum_k count(s_{ik})}{N}) \qquad (3.1)$$

where c_i is the concept conveyed by word w_i, s_{ik} are all the words subsumed by c_i and N is the overall number of noun or verb or adjective/adverb tokens depending on whether w_i is a noun or a verb or a adjective/adverb respectively. ICC stands for Information Content of Concepts.

To compute ICC, a corpus and a taxonomy of concepts are necessary, Resnick uses WordNet (Fellbaum (1998), see section 5.1.2 for more details). Figure 3.1 shows a fragment of WordNet where the concepts of "dime" and "nickel" are subsumed by the more generic concepts of "coin" and "cash" so when computing ICC any occurrence in the corpus of the noun "dime" is counted for the ICC of "dime", "cash" and "coin". Note that no word sense disambiguation is applied when counting.

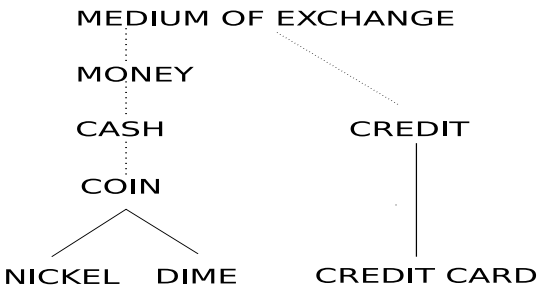

Figure 3.1: *A fragment of the WordNet taxonomy (from Resnick (1995)). "Dime", "nickel" and "credit card" are the most specific concepts in the fragment while "medium of exchange" is the most general.*

[1]In Resnick (1995) this measure is called Information Content, but in order to distinguish it from Pan and McKeown (1999)'s Information Content we call it Information Content on Concepts (ICC).

Because of semantic ambiguity a word can be associated to different concepts (i.e., senses) and so to different ICC's. As a consequence, we heuristically chose the minimum ICC (which is the ICC of the most frequent sense of a given word).

Finally, since ICC is a measure of informativeness complementary to IC, both measures have been used as predictive features[2].

ICC was extracted using the WordNet::Similarity Perl module (Pedersen et al. (2004)).

3.2.2 Syntactic Dependencies (SD)

Along with POS other deeper syntactic features have been proposed to account for syntactic effects on pitch accenting. Their utility in accent prediction is not clear. Pan et al. (2002) report that the use of syntactic constituents and functions (e.g., subject, object, etc...) do not help improving accent prediction, while Sridhar and Bangalore (2008) show the utility of supertags, syntactic tags that encode predicate-argument information. The two works use different corpora.

Here we propose the use of dependency grammars to account for some possible syntactic effects and effects of relative informativeness on pitch accenting.

In a syntactic dependency grammar the syntax is described by binary relations (i.e., dependencies). Figure 3.2 shows a dependency tree for the sentence "The musician in red shirt only plays Scarlatti" where relations are graphically represented by edges (including direction). In the example the words "musician" and "plays" are linked by the relation subject-of (i.e., Sbj-of(musician,plays)), "red" and "shirt" by noun-modifier-of, etc... In each relation one word is the head and the other is the dependent. For example in Sbj-of(musician,plays) "musician" and "plays" are *dependent* and *head* respectively.

To identify syntactic dependencies in text we used the Malt syntactic dependency parser (Nivre et al. (2007)) and then extracted the following features for any word of the parsed sentence:

1. Relation's name of the relation of which the current word is a dependent (e.g., Sbj-of for word "musician"). Note that a word can be a dependent at most once.

2. Path in terms of edges (and their directions) between the current word and the next word (e.g., Obj-of$\downarrow$ for word *play*, with the up-down arrow indicating that a

[2] An alternative solution could be that of interpolating them, i.e., using a new feature that would be a weighted sum of ICC and IC.

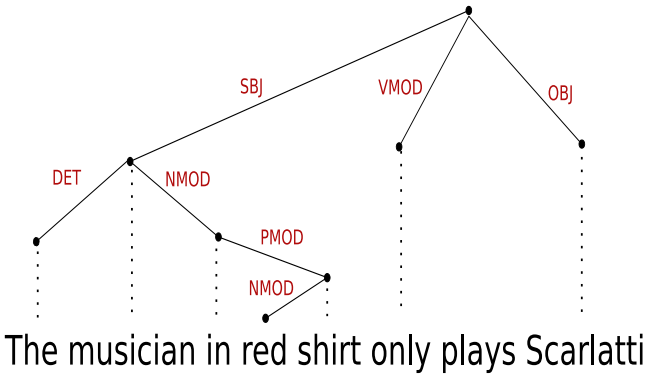

Figure 3.2: *An example of dependency tree. Dependency tree of the sentence "The musician in red shirt only plays Scarlatti"*

step down in the dependency tree is made when moving from *plays* to *Scarlatti*).

3. Path in terms of directions of relations between the current word and the next word (e.g., Relation↓-Relation↓ for word "in", the names of the relations are not specified).

4. Path length between the current word and the next word in number of edges.

In feature 2 if the path is not a one-edge only (e.g., Sbj-of↑) or a 2-edge "up-and-down" path (e.g., Relation↑-Relation↓), the path value is collapsed to a "Long" value. That is a heuristic to avoid data sparsity. It is motivated by the fact that the 1-edge and the 2-edge "up-and-down" paths only occur between siblings (2-edge "up-and-down") and parent-children (1-edge). All other paths only occur for more distant relationships.

In feature 3 when a path is longer than 2 edges it is mapped into a "Long" value.

Feature 1 is mainly intended to account for a possible correlation between accenting and syntactic function. Features 2 and 3 are intended to account for a possible mapping between the syntactic relation linking two consecutive words and their prosodic prominence relation (which in turn might affect the accenting of the words). Finally

features 2, 3 and 4 are also intended to implicitly convey information about prosodic structure, which is highly correlated with syntactic structure and which in turn might affect accent placing.

3.2.3 Dependency-based Relative Informativeness (DRI)

Syntactic dependencies are also used to build a "Relational" Language Model. The "Relational" Language Model is intended to be complementary to the probability of bigrams in the quantification of relative informativeness.

We define a "Relational" Language Model as a bigram model in which the probability of a word w_i is conditioned on the closest single-linked (to w_i) word preceding w_i. Formally:

$$p(w_i|w_k, D(w_i, w_k) = 1) \tag{3.2}$$

where $D(w_i, w_k) = 1$ when there is a dependency linking the two words and w_k is the closest single-linked (to w_i) word preceding w_i ($k < i$).

To train the "Relational" Language Model we first ran the Minipar syntactic dependency parser Lin (1998a) on the same corpus used to estimate probability of unigrams and of bigrams, and for each word w_i we extracted the (w_i, w_k) pair satisfying $D(w_i, w_k) = 1$ AND $k < i$. Each word pairs was put on a single line as it were a whole sentence. The resulting corpus was then used to train a 2-gram language model using the SRI Language Model Toolkit (Stolcke (2002)).

In order to have an information-theoretic measure we used the negative of the logarithm of the conditional probability:

$$DRI(w_i) = -\log(p(w_i|w_k, D(w_i, w_k) = 1)) \tag{3.3}$$

where DRI stands for Dependency-based Relative Informativeness.

3.2.4 Cache Information Content (CIC)

We have seen in section 2.4 that features accounting for the effect of the new/given dichotomy on pitch accenting do not seem to be very effective. Only Hirschberg (1993) shows some accuracy improvement when using this kind of features, although disappointingly smaller than that expected by the author.

Contrary to previous work in which new/given features are binary or at most a three-value feature indicating the information status independently of their intrinsic

informativeness, we look at information status as at a discourse-context effect on the (information-theoretic) informativeness of words. To do that we use a Cache Language Model (Kuhn and De Mori (1990)).

In a Cache Language Model the probability of a word is raised if the word previously occurred in a segment of text (e.g., if it occurred in the previous 200 words). This dynamic word probability is computed by statically interpolating the word probability from the standard Language Model (e.g., the 9 million tokens corpus from The Herald news) and the word probability in a cache containing the last N seen words (Cache Language Model). Formally:

$$p(w_i) = \mu p(w_i|LM) + (1-\mu)p(w_i|CLM) \tag{3.4}$$

where LM is the standard Language Model, CLM is the Cache Language Model, and μ is a constant. In all experiments μ was manually set to 0.1 and the cache size to 100 words[3]. The cache is never flushed.

The Cache Information Content is simply the negative logarithm of this dynamically updated unigram probability. It can be seen as another measure of relative informativeness where the word informativeness is not (or not only) affected by sentence-context but by the wider discourse context.

CIC was estimated using the SRI Language Model toolkit (Stolcke (2002)) (see Weintraub (1995) for details on the computation of a Cache Language Model).

3.2.5 Normalisation (NZ)

When looking at the errors made by a first implementation of our accent predictor it was immediately clear that contractions (e.g., don't, it's, etc...) were over accented either because the POS tagger did not always correctly tag them or because they had high IC. The high IC was due to the fact that there are no contractions in the (The Herald news) corpus used to train the language models.

To solve this problem, contractions were expanded in their non-contracted counterpart (e.g., don't → do not) and features were extracted on the resulting normalised text. A feature was used to indicate those words that were originally in a contracted form. After prediction, originally contracted words were contracted again and the accent value was assigned to the contracted word according to some lexical rules. Also other minor normalisations were done.

[3]The low value of μ aims to differentiate CIC and IC as much as possible

We mention these technical details here as they turned out to be very effective , more than some of the "fancier" features described above. The normalisation has been done for all the predictors evaluated (see section 3.5).

3.3 Machine Learning Techniques

The choice of the machine learning technique certainly has an impact on the accuracy of the pitch accent predictor. However a grid search for the most suitable technique among a very large set of available techniques is not a good idea. The comparison of predictors based on so many techniques would be dependent on so many variables (e.g., the training and testing data used, the parameters of each technique, the implementation of each technique) that its results may be either unfeasible or unreliable.

In previous work two different families of machine techniques have been used for pitch accent prediction: one implies that the accenting of the current word does not depend on the accenting of the previous words (see Yuan et al. (2005) for example), while the other does not imply this independence (see Gregory and Altun (2004) and Levow (2008) for example).

Formally, in the first family the conditional probability $p(y_1, \cdots, y_n, |\mathbf{x_1}, \cdots, \mathbf{x_n})$, where y_i is the accent value (i.e., $\pm$accent) on word w_i and $\mathbf{x_i}$ is the vector of training features on word w_i, is:

$$p(y_1, ..., y_n, |\mathbf{x_1}, ..., \mathbf{x_n}) \propto \prod_{i}^{N} \Phi(y_i, \mathbf{x_{i-M}^{i+M}}) \tag{3.5}$$

where $\mathbf{x_{i-M}^{i+M}}$ consists of $\mathbf{x_i}$ and all the M $\mathbf{x}$ vectors preceding and following it ($\mathbf{x_i}$ is usually called the observation window, if $\mathbf{x_i}$ is observable), while Φ is a generic function.

In the the second family:

$$p(y_1, ..., y_n, |\mathbf{x_1}, ..., \mathbf{x_n}) \propto \prod_{i}^{N} \Phi(y_i, \mathbf{x_{i-M}^{i+M}}, \mathbf{y_{i-L}^{i-1}}) \tag{3.6}$$

where $\mathbf{y_{i-L}^{i-1}}$ is the vector of the L accent values preceding y_i.

Although classifiers from both families have been proposed there is surprisingly practically no experimental work investigating which of the two families is most appropriate for pitch accent prediction. Such a comparison would spread some light on to which extent the knowledge of the accentuation history can improve the modelling of sequences of pitch accents.

In order to carry out such comparison we used four different machine learning techniques: Classification And Regression Trees (Breiman et al. (1984),Quinlan (1993)), Bagging (Breiman (1996)), Hidden Markov Models, and Conditional Random Fields (Laffeerty et al. (2001)). The first two techniques imply an independence assumption, the last two do not.

In all these methods the predicted sequence $y_1,...,y_n$ is the one that maximises $p(y_1,...,y_n,|\mathbf{x_1},...,\mathbf{x_n})$.

The following sections briefly describe these four techniques. Since the accent prediction is a classification task we only look at these techniques as classifiers.

3.3.1 Classification And Regression Tree (CART)

The Classification And Regression Tree technique classifies data points by searching for a "good" decision tree. A decision tree is a binary tree that groups the training data points at its leaf nodes, each of them associated to a classification value. A leaf node can contain data points having different classification value, the classification value associated to a leaf node is the the majority classification values. Each non-terminal node of the tree is a test on the value of some training feature of the data points. During prediction a new data point is associated to a leaf node (and so classified) by starting from the root node, doing the test associated to that node and then moving down the branch specified by the result of the test, doing the test at the node at the end of the branch, and so on until a leaf node is reached. Figure 3.3 shows an example of a decision tree.

A "good" decision tree is a tree that both fits the training data points well and generalises well. To fit the training data points well its leaf nodes should be as much uniform as possible, i.e., they should contain a large majority of data points having the same class (in information-theoretic terms they should have a low entropy on the classification variable). However trying to fit the training data as best as possible would lead to overfitting (in which in the extreme case each leaf node would contain a single data point), i.e., the predictor would have a very high accuracy on the training data but a poor performance on the testing data as a consequence of its poor generalisation capability.

A decision tree is built[4] by first searching for the test (node) that best splits the training data set into two subsets, then, for each subset, the test (node) that best splits

[4]The creation of a decision tree can also be seen as a search in the space of all decision trees available.

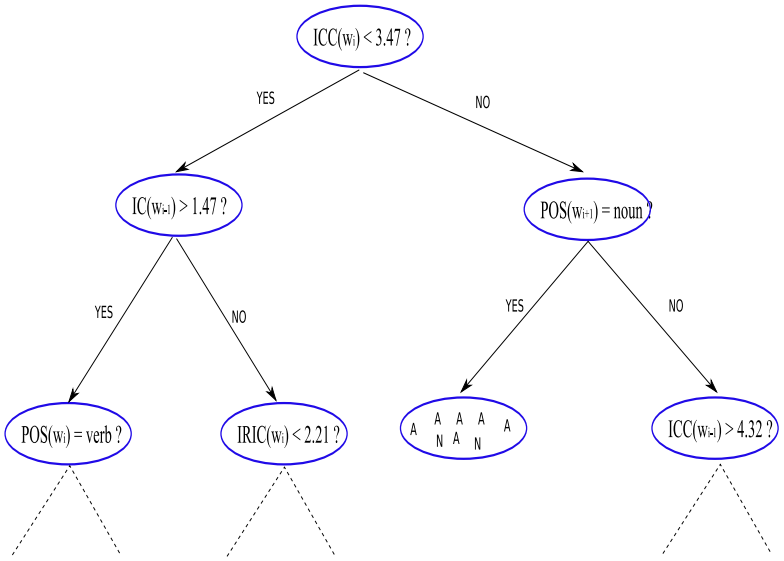

Figure 3.3: *Fragment of a decision tree. Each non-terminal node is associated with a question on some feature's value. The only terminal node (the third from the left on the bottom) groups some data points. It is associated with the classification value A as it contains more A's than N's. A and N stand for accent and no-accent respectively*

it into two further subsets is selected, and so on until the "good" tree is completed. Intuitively a node splits well a set into two new subsets if the distributions of the classification value in the two subsets is more uniform (i.e., has a lower entropy) than in the original subset. Information-theoretic measures are used to compute the goodness of a split. Note that each optimal split is selected at a local level, i.e., node level, so there is no guarantee that the sequence of splits, and so the resulting tree, is globally optimal (i.e., creates the most uniform possible subsets).

When predicting the class y_i (i.e., finding the leaf node) of a new data point also the probability $p_T(y_i|\mathbf{x_i})$ of the prediction is given (where $\mathbf{x_i}$ is the vector of training features for data point i). Such probability is given by the number of training data points in the leaf node having the majority class divided by the overall number of training data points in the leaf node.

Going back to the pitch accent prediction task and equations 3.5 and 3.6 when using CART the conditional probability $p(y_1,...,y_n,|\mathbf{x_1},...,\mathbf{x_n})$ is approximated as follows:

$$p(y_1,...,y_n,|\mathbf{x_1},...,\mathbf{x_n}) = \prod_i^N p_T(y_i|\mathbf{x_{i-M}^{i+M}}) \tag{3.7}$$

Finding the $y_1,...,y_n$ that maximises $p(y_1,...,y_n,|\mathbf{x_1},...,\mathbf{x_n})$ simply means finding at each word w_i the y_i that maximises $p_T(y_i|\mathbf{x_{i-M}^{i+M}})$.

Note that $p_T(y|\mathbf{x})$ is different from $p(y|\mathbf{x})$, that is the probability computed by simply counting for each value of $\mathbf{x}$ the frequency of the values of y. However $p_T(y|\mathbf{x})$ can be seen as a smoothed estimation of $p(y|\mathbf{x})$ in that the number of 0-counts is largely reduced as different vectors (i.e., different values of $\mathbf{x}$) are grouped in the same leaf node and share the same conditional probability.

We used two different implementations of CART: Wagon from the Edinburgh Speech Tools (Taylor et al. (1999)) and J48 from Weka (Hall et al. (2009)).

3.3.2 Bagging

Bagging (which stands for bootstrap aggregating) is a method that uses a committee of predictors (e.g., a committee of CART's) to generate an aggregate prediction. Given a training data set $\mathcal{T}$ consisting of N data points, Bagging first creates L bootstrap data sets. A bootstrap data set $\mathcal{T}_\mathcal{B}$ is created by randomly drawing N data points from $\mathcal{T}$ with replacement so that some data points of $\mathcal{T}$ may replicate in $\mathcal{T}_\mathcal{B}$ and some other may not occur.

Once the bootstrap data sets have been created a predictor (e.g., a CART) is trained on each of them so that a committee of L predictors is created. When Bagging classifies new data points, it classifies by (majority) voting.

Bagging could be potentially used with any machine learning algorithm. However it has been shown, both theoretically and empirically in Breiman (1996), to be able to improve the prediction accuracy (with respect to a single predictor trained on $\mathcal{T}$) only when the algorithm is unstable. An algorithm is unstable when small changes in the training data set (in terms of replacement, not of training features) cause large changes in the prediction[5].

CART is an unstable predictor and so one obvious way to improve its accuracy in the pitch accent prediction task is to use it within Bagging. Another motivation to use Bagging with CART is that, because CART instability, the error analysis at the

[5]Somehow the instability of a predictor can also be seen has a tendency to overfit since its prediction is "stuck" to the training data set

sentence level of a CART pitch accent predictor may not be reliable[6].

The Bagging implementation we used is a Weka implementation using the j48 CART.

3.3.3 Hidden Markov Models with "CART estimation of emission probabilities" (CART-HMMs)

In a HMM approach to accent prediction we can look at each y_i as a hidden state which emits the observation state (which is an observation vector) $\mathbf{x_i}$ with probability $p(\mathbf{x_i}|y_i)$[7]. The probability of moving from state y_{i-1} to state y_i after having moved from state y_{i-L} to state y_{i-1} is given by the transition probability $p(y_i|y_{i-L}^{i-1})$.

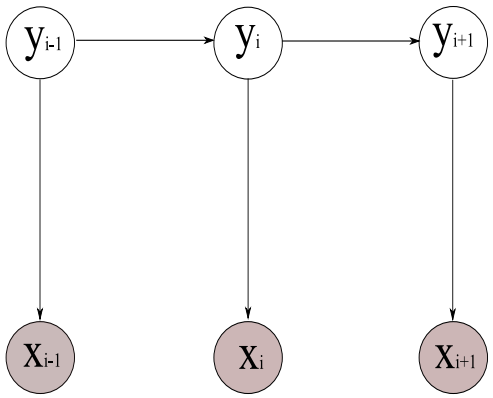

Figure 3.4: *Graphic representation of a 1st order HMM. The filled nodes are the observed variables while the other nodes are the hidden variables. The edges from the hidden nodes to the observed nodes represent the emission probabilities, while the edges between hidden nodes represent the transition probabilities.*

In HMMs the joint probability $p(y_1,...,y_n,\mathbf{x_1},...,\mathbf{x_n})$ can be approximated as follows:

$$p(y_1,...,y_n,\mathbf{x_1},...,\mathbf{x_n}) = \prod_{i=1}^{n} p(\mathbf{x_i}|y_i)p(y_i|y_{i-L}^{i-1}) \tag{3.8}$$

[6]For example if during training we want to change the size of the held-out set used for CART pruning (which serves to limit overfitting) we may have quite different predictions.

[7]Note that here we are considering the vector $\mathbf{x_i}$ instead of $\mathbf{x_{i-M}^{i+M}}$. However the observation vector at state i could also include features of the preceding and following words.

where L is the order of the HMMs (e.g., a first order Hidden Markov model has $L = 1$, which means $p(y_i|y_{i-L}^{i-1}) = p(y_i|y_{i-1})$). A graphic representation of a first order HMM is shown in figure 3.4. Note that when using HMMs for pitch accent prediction there is actually no hidden variable. That simplifies the estimation of both emission and transition probabilities.

HMM decoding, i.e., looking for the sequence $(y_1, ..., y_n)$ that maximises $p(y_1, ..., y_n|\mathbf{x_1}, ..., \mathbf{x_n})$ is equivalent to look for the sequence that maximises the joint probability $p(y_1, ..., y_n, \mathbf{x_1}, ..., \mathbf{x_n})$:

$$\arg\max_{y_1,...,y_n} p(y_1, ..., y_n|\mathbf{x_1}, ..., \mathbf{x_n}) = \arg\max_{y_1,...,y_n} \frac{p(y_1, ..., y_n, \mathbf{x_1}, ..., \mathbf{x_n})}{p(\mathbf{x_1}, ..., \mathbf{x_n})} = \arg\max_{y_1,...,y_n} = p(y_1, ..., y_n, \mathbf{x_1}, ..., \mathbf{x_n}) \tag{3.9}$$

where $p(\mathbf{x_1}, ..., \mathbf{x_n})$ does not affect the maximization.

The transition probabilities are simply estimated by frequency counting on the training data. Concerning the emission probabilities, by using the Bayesian rule probability $p(\mathbf{x_i}|y_i)$ can be expressed as follows:

$$p(\mathbf{x_i}|y_i) = \frac{p(y_i|\mathbf{x_i})p(\mathbf{x_i})}{p(y_i)} \tag{3.10}$$

Again $p(\mathbf{x_i})$ can be ignored as we are only interested in $\arg\max_{y_1,...,y_n} p(y_1, ..., y_n|\mathbf{x_1}, ..., \mathbf{x_n})$.

$p(y_i)$ is again estimated by frequency counting while for $p(y_i|\mathbf{x_i})$ we used the "CART estimation" proposed by Sun and Applebaum (2001).

The implementation of CART we used is Wagon (with a 3-word observation window), while for the HMM part we wrote our own implementation that includes both first order (i.e., $L = 1$) and second order (i.e., $L = 2$) Markov models. In our implementation we introduced a new parameter λ (with $0 \leq \lambda \leq 1$) that modifies equation 3.8 as follows:

$$p(y_1, ..., y_n, \mathbf{x_1}, ..., \mathbf{x_n}) = \prod_{i=1}^{n} (p(\mathbf{x_i}|y_i))^{\lambda} (p(y_i|y_{i-L}^{i-1}))^{(1-\lambda)} \tag{3.11}$$

The λ parameter allows to assign different weights to the emission and the transition probabilities respectively. Its use is motivated by the fact that one of the two probabilities may be less important than the other in the modelling of the joint probability. For example if the accenting of a word is only weakly influenced by the placement of the previous accents then we can set λ to a high value to account for that.

Note that when $y = accent$ and $y = noaccent$ are equiprobable, setting $\lambda = 1$ is equivalent to using CART.

To decode the HMM, i.e., to find $\arg\max_{y_1,...,y_n} p(y_1,...,y_n|\mathbf{x_1},...,\mathbf{x_n})$, we used Viterbi decoding for the first-order HMM and $A*$ decoding for the second order HMM[8].

3.3.4 Conditional Random Fields (CRFs)

Instead of modelling the joint probability $p(y_1,...,y_n,\mathbf{x_1},...,\mathbf{x_n})$ to find the sequence that maximises the conditional probability $p(y_1,...,y_n|\mathbf{x_1}...,\mathbf{x_n})$ we can directly model the conditional probability. This is the main difference between HMMs and Conditional Random Fields and that makes the former belong to the family of the generative methods and the latter to the family of the probabilistic discriminative approaches[9].

"A Conditional Random Field is simply a conditional distribution $p(y_1,...,y_n|\mathbf{x_1}...,\mathbf{x_n})$ with an associated graphical structure" (Sutton and McCallum (2007)). For sake of simplicity here we only take into account a special case of CRFs, the linear chain CRF.

In a linear chain CRF the conditional probability is approximated as follows:

$$p(y_1,...,y_n|\mathbf{x_1},...,\mathbf{x_n}) = \frac{1}{Z(\mathbf{x_1},\cdots,\mathbf{x_n})} \exp\{\sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k f_k(y_i,y_{i-1},\mathbf{x_i})\} \tag{3.12}$$

where $Z(\mathbf{x_1},\cdots,\mathbf{x_n})$ is an instance-specific normalisation function, λ_k are weights (whose values are computed during training) and f_k are indicator functions. For example a possible f_k for a CRF-based pitch accent predictor could be:

$$f_k = \begin{cases} 1 & if\ y_i = accent\ and\ x_{ik} = POS(w_i) = verb \\ 0 & otherwise \end{cases} \tag{3.13}$$

where x_{ik} is the value of the feature x_k of the feature vector $\mathbf{x_i}$. This set of f_k of this type serves the same function of the emission probability in HMMs.

While the set of functions of this type:

$$f_k = \begin{cases} 1 & if\ y_i = accent\ and\ y_{i-1} = no-accent \\ 0 & otherwise \end{cases} \tag{3.14}$$

is comparable to the transition probability of HMMs. Note that in a CRF we can additionally use functions of the following type (as we actually did in our implementation):

$$f_k = \begin{cases} 1 & if\ y_i = accent\ and\ y_{i-1} = no-accent\ and\ x_i = POS(w_{i_i}) = noun \\ 0 & otherwise \end{cases} \tag{3.15}$$

[8]When using a second order HMM the Viterbi decoding can not be used as the Viterbi search (of the best sequence of hidden states among all the possible sequences) requires that the probability of a hidden state only depends on the previous state. The $A*$ search, which is a kind of *best-first* search, allows to decode HMMs of second or higher order.

[9]Making an analogy to methods for non-sequential data HMM is the equivalent to Naive Bayes for sequential data while CRF is the equivalent of logistic regression

The implementation used is FlexCRFs (Phan et al. (2005)) which allows the use of first order and second order (in which $f_k(y_{i-2}^i, \mathbf{x_i})$) CRFs. Since this implementation, like almost all the other implementations of CRFs, does not deal with continuous valued features, we discretized the training features into N bins where N was predetermined. The set of features used (i.e., $\mathbf{x}$) in the CRF-based accent predictor is the same of that used in CART and HMM-based predictors. The set of f_k indicator functions is obtained by combining all possible values of y_i, y_{i-1} and x_i.

3.4 Data

3.4.1 The Boston University Radio News Corpus

The Boston University Radio News Corpus (BURNC) (Ostendorf et al. (1995)) is a corpus of American English read speech. It consists of seven hours of speech recorded from seven speakers (f1a,f2b,f3a,m1b,m2b,m3b,mb4) while reading radio news. The speech of six speakers (mb4 is excluded) is annotated according to ToBI conventions.

One part of the corpus consists of the recorded speech of only one speaker (f2b) and contains 9473 words (tokens). This part was used for the training and testing of the pitch accent predictors presented here.

The other part of the corpus consists of speech from all the six speakers reading the same text. We used this part to analyse the variability in pitch accent placements which is addressed in the next chapter. This part was also used to evaluate the accent predictor on a "multi-speaker" data set. Results of this evaluation are shown and discussed in the next chapter.

3.4.2 The Switchboard Corpus

The Switchboard Corpus (SWBDC) (Godfrey et al. (1992)) is a corpus of America English spontaneous speech. It consists of 2430 telephonic conversations. Subsets of the Switchboard have been prosodically annotated using (simplified) ToBI conventions by Ostendorf et al. (2001) and Calhoun (2006). We used a subset of the subset annotated by Calhoun (2006) that consists of 19231 tokens.

3.5 Results

The accuracy predictions shown in this section have been computed using a 10-fold cross validation in the training data set. For both BURNC and SWBDC we have used two different types of data set, one including punctuation marks and one without punctuation marks. Most of the results shown here have been computed on the version with punctuation marks. However for a fair comparison with predictors from previous work in which punctuation marks are usually not included we also show the accuracy of our best predictor on the version without punctuation marks.

In BURNC the punctuation marks were already available (as the speech is read speech) while in SWBDC we inserted a full stop at each change of turn (i.e., at the point where a speaker stops speaking and the other starts speaking). The BURNC version with punctuation marks consists of 10457 tokens (9473 words + 984 punctuation marks), while SWBDC version with punctuation marks consists of 20000 tokens (19231 words + 769 punctuation marks).

3.5.1 Results on the Boston University Radio News Corpus

Table 3.1 compares the prediction accuracy of predictors based on the four machine learning techniques. The set of training features used for this comparison does not contain all the training features but only IC, RIC, POS (and "normalisation features" (section 3.2.5), as for all the predictors in this section). We only used these features as the comparison between machine learning techniques was carried out before the extraction of the new features (and once the best technique was identified we used it to compare the training features). We do not expect the use of the full feature set to produce different results.

All the predictors used for evaluation used a 3-word observation window. The Bagging-based predictor turned out to be the best predictor, immediately followed by CART in the Weka implementation. The large difference between the two different implementations of CART is primarily due to different default settings for tree pruning (there is actually no pruning by default in Wagon).

Surprisingly both HMM (with $\lambda = 0.5$) and CRF-based predictors performed worse than CART and Bagging-based predictors. Such result would suggest that, at least on this speaker-dependent corpus, the independence assumption leads to better results.

However to further investigate whether the knowledge of previous accent values has some utility in the prediction of accent patterns the first order HMM-based pre-

Method	Accuracy
CART (Wagon)	84.93%
CART (Weka)	85.96%
Bagging	86.02%
1st order CART-HMM	84.43%
2nd order CART-HMM	83.48%
2nd order CRF	83.01%

Table 3.1: *Machine learning techniques and accent prediction accuracy. The predictors are evaluated on the BURN corpus. The training features used are IC, RIC, POS and "normalisation features".*

dictor was evaluated using different values of λ (see equation 3.11). Note that in f2b the number of accented words is almost identical to that of non-accented words so the HMM-CART having $\lambda = 1$ is equivalent to CART. The results are shown in table 3.2

As expected a small value of λ, which corresponds to a higher weight on $p(y_i|y_{i_1})$ than on $p(y_i|\mathbf{x_i})$, causes a decrease of accuracy. On the other hand a high value of λ improves the accuracy with respect to a "standard" HMM-based predictor. The accuracy is even higher than that of the CART(Wagon)-based predictor (85.15% vs. 84.93%, remember that the CART-HMM predictor is based on the Wagon implementation of CART) although the difference is very small. This result suggests that, for the purpose of accent prediction, the information coming from emission probabilities is more relevant than that coming from transition probabilities.

It is possible that if $p(y_i|\mathbf{x_i})$ is estimated using Bagging (of CARTs) the performance of the HMM based predictor is slightly better than that of a Bagging predictor (although with an improved estimation of $p(y_i|\mathbf{x_i})$ from Bagging the importance of $p(y_i|y_{i-1})$ may decrease).

Finally, note that the λ value is not estimated on the training data by maximum likelihood but it is manually set, so a "fair" use of λ would require a maximum likelihood estimation or a search for the best value of λ on a validation set[10].

In order to investigate the importance of the new training features proposed here, a Bagging-based predictor was trained and evaluated using different sets of features.

[10]Actually the validation data should be used anytime we need to set the parameters of the machine learning methods. To avoid that we have used the default values of the parameters

λ value	Accuracy
0.2	79.5%
0.5	84.43%
0.8	85.15%

Table 3.2: *Changing λ in the HMM predictor. The impact of λ on prediction accuracy.*

Feature Set	Accuracy
Old	86.25%
Old + ICC	86.79%
Old + SD	86.57%
Old + DRI	86.17%
Old + CIC	86.17%
All	86.85%

Table 3.3: *Using different features set in accent prediction. Evaluation on the BURN corpus.*

The accuracy of these predictors is shown in table 3.3[11]. Note again that normalisation is included in each predictor.

The ICC feature is the most predictive among the new features. The features extracted from the dependency syntactic parser increase accuracy, although the accuracy increase is less the those due to ICC.

On the other hand features DRI and CIC do not improve accuracy. However the best combination of new features is the combination including all the new features. The predictor using all (new + old) features gives a 86.85% accuracy.

In order to see the correlation of each feature (both new and old features) to pitch accenting, the symmetric uncertainty (Witten and Eibe (2005)) between each feature and the accent value is computed. The symmetric uncertainty is a normalised version

[11]Note that the prediction accuracy of the Bagging-predictor using old features only is different from the one shown in table 3.1. That is due to the use of the stratified cross-validation provided by Weka (it is the default cross-validation). The stratified cross-validation preserves in both training and testing set the ratio of accents/no-accents in the whole data set. It can not be used when testing the HMM and CRF predictor because it does not necessarily preserve the order of the accents.

Rank	Symmetrical Uncertainty	Feature
1	0.266	ICC
2	0.238	IRIC
3	0.238	IC
4	0.215	CIC
5	0.184	WL

Table 3.4: *Features correlation with the accent class. The table shows the five most correlated features on the BURN corpus*

Punctuation	Window Size	Accuracy
YES	1	86.26%
YES	3	86.85%
YES	5	86.85%
NO	3	85.2%

Table 3.5: *Effect of observation window size and punctuation on prediction accuracy of the best predictor (i.e., Bagging-based predictor). Evaluation on the BURN corpus.*

of the mutual information between two variables[12]. The symmetric uncertainty values for the five most correlated features are shown in table 3.4. Note that the correlation between a feature and pitch accenting does not say how much useful a feature actually is when combined with the other features but only how much useful it is when used alone.

Finally table 3.5 shows the accuracy of the best predictor (Bagging-based) on the data set with and without punctuation marks and with different observation window sizes. The 3-word window gives the best result.

[12]The symmetric uncertainty is defined as $2\frac{I(x;y)}{H(x)+H(y)}$, where x and y are two random variables, $I(x;y)$ is the mutual information between the two variables, $H(x)$ and $H(y)$ are the entropies of the two variables.

3.5.2 Results on the Switchboard Corpus

The features extracted to train the different predictors on SWBDC have been extracted in the same way they have been extracted from BURNC except for a difference in the computation of CIC. While for the prediction on BURNC CIC was computed interpolating the cache language model with the "standard" language model (i.e., the language model trained on 9 million words of the Herald News), for the prediction on SWBDC the CIC was computed interpolating the cache language model with a language model resulting from the dynamic interpolation of the "standard" language model with a language model trained on the whole SWBD corpus[13]. In the dynamic interpolation the probability of the unigram is computed as follows:

$$p(w_i) = \mu(w_i)p(w_i|LM1) + (1-\mu(w_i))p(w_i|LM2) \tag{3.16}$$

where $\mu(w)$ is the posterior probability of "being" in LM1 given word w_i[14](see Weintraub (1995) for more details).

The dynamic interpolation of the "standard" language model with the language model trained on SWBDC was carried out to: 1) better model spoken language; and 2) to compensate for the fact that a language model only trained on a small corpus like SWBDC has too many zero-count bigrams.

Table 3.6 shows the prediction accuracy of predictors based on different machine learning techniques. As for the evaluation on BURNC the only features used are IC, RIC, and POS.

On this corpus the prediction accuracies are very close to each other. CART in the Weka implementation turned out to be the best predictor. However, since the second order HMM-CART uses Wagon CART for the estimation of $p(y_i|\mathbf{x_i})$ and it performs better than Wagon CART we expect a HMM-CART using Weka CART to perform better than the Weka CART. In other words, on SWBDC the information about the previous accent values seems to be more important than on BURNC. Concerning the λ parameter in CART-HMM the 0.5 value in the second order HMM gives the best result. The impact of each new feature on accent prediction accuracy is shown in table 3.7. A

[13]The interpolation could also be done to compute IC but at the moment IC is computed using the CMU Language Model Toolkit which does not allow dynamic interpolation. However we also wanted to compute IC as in previous work, that is with no dynamic interpolation of any kind.

[14]In the SRI Language Model Toolkit this posterior probability is actually a weighted posterior probability in which one Language Model can have more weight than the other one. For the pitch accent predictor the Language Model trained on SWBD was given a much higher weight (0.9) than the standard Language Model

Method	Accuracy
CART (Wagon)	74.79%
CART (Weka)	75.17%
Bagging	74.73%
1st ord. CART-HMM	74.69%
2nd ord. CART-HMM	75.01%
2nd ord. CRF	74.51%

Table 3.6: *Machine learning techniques and accent prediction accuracy on SWBDC. The predictors are evaluated on the Switchboard corpus. The training features used are IC, RIC, POS and "normalisation features".*

Feature Set	Accuracy
Old	77.27%
Old + ICC	77.53%
Old + SD	77.01%
Old + DRI	77.25%
Old + CIC	77.72
All	77.7%

Table 3.7: *Using different features set in accent prediction Evaluation on the SWDBC corpus.*

Bagging-based predictor was used[15]. The ICC feature increases accuracy, consistently with results on BURNC. The CIC features turned out to be the most predictive and when combined with the old features produces better results than the whole set of features.

The symmetrical uncertainty was used again to have a measure of the correlation of each feature with pitch accenting. Table 3.8 shows the five most correlated features. The top-five features on SWBDC and on BURNC are the same, although not in the same order.

Finally table 3.9 shows the accuracy of the best predictor (Bagging-based) on the

[15]We used Bagging instead of CART as on the Weka stratified cross validation Bagging has a slightly better accuracy.

Rank	Symmetrical Uncertainty	Feature
1	0.09	ICC
2	0.079	IC
3	0.071	IRIC
4	0.071	WL
5	0.066	CIC

Table 3.8: *Features correlation with the accent class. The table shows the five most correlated features on the Switchboard corpus*

Punctuation	Window Size	Accuracy
YES	1	76.76%
YES	3	77.7%
YES	5	77.7%
NO	3	75.84%

Table 3.9: *Effect of observation window size and punctuation on accuracy of the Bagging-based predictor. Evaluation on the Switchboard corpus*

data set with and without punctuation mark and with different observation window sizes.

The 3-word window and the 5-word window give the best results.

3.6 Discussion

In this chapter two issues have been addressed. The first issue concerns the interdependence of pitch accent placements. Two different types of machine learning techniques were compared to investigate whether assuming accent placements interdependence leads to a better modelling of pitch accent patterns. Results show that the knowledge of the previous accent values has a small (especially in read speech, see table 3.2) impact on the accuracy of accent predictors.

Note that such results do not mean that the accent values of preceding words has a very small effect on the accenting of the current word but they only imply that the machine learning algorithms that model accent placements interdependence are not

good enough in modelling such interdependence[16].

To improve such modelling in the HMM-based predictor we have introduced a new parameter (λ) to weight the relative importance of emission and transition probabilities. This produced (in read speech only) an improvement over the "standard" HMM-based predictor (i.e., HMM with equal weights on transition and emission probabilities) from 84.43% to 85.15% but only a very small improvement (from 84.93% to 85.15%) over the 0-order HMM-based predictor, i.e., the CART-based predictor.

Comparing the two algorithms that assume accent interdependence the HMM-based predictors always outperformed the CRF-based predictor. This result contrasts with the results shown in Gregory and Altun (2004) where a CRF-based predictor outperformed an HMM-based predictor. The CRF predictor used here and that used in Gregory and Altun (2004) seem to be very similar so the difference of results should reside in a difference in the HMM-based predictors, with the main (and perhaps only) difference being in the estimation of $p(y_i|\mathbf{x_i})$, with the CART-based estimation being a better estimation than that used in Gregory and Altun (2004) (which is not described in the paper). Finally a partial justification for the CRF predictor poor performance might reside in the fact that in the CRF predictor the continuous-valued features have been binned into equal categories (as in Gregory and Altun (2004) with the same number of bins). When increasing the number of bins the accuracy slightly improved, so perhaps a smarter discretization of the continuous-valued features (e.g., the discretization proposed by Fayyad and Irani (1993), which is the same used in Weka-CART) could lead to better results.

The second issue addressed in this chapter concerns the utility of the training features proposed here. Results show that the proposed features increase the predictor's accuracy from 86.25% to 86.85% on BURNC and from 77.2% to 77.7% on SWBDC. The ICC (Information Content of Concept) feature is the most useful feature on BURNC while CIC (Cache Information Content) is the most useful on SWBDC. The features extracted from the syntactic dependency parser have some utility on BURNC but not on SWBDC. A possible explanation is that the dependency parser is much less accurate on SWBDC than on the BURNC as SWBDC is more "ungrammatical".

[16]This claim is supported by the results shown by Brenier et al. (2006) where the knowledge of the real accent-values of the surrounding words is shown to significantly improve prediction. Note that the CART-based predictor used in Brenier et al. (2006) (named "FS.CONTEXT+ORACLE") uses oracle features, that is the real accent-values of the previous and next word as features, in both the training and testing phase. As a consequence it can not be used for a real prediction task in that in a real prediction task the values of the oracle features are unknown (they are the values the predictor has to find out).

In general the information-theoretic features measuring the informativeness of words (especially intrinsic informativeness but also relative) are the most predictive features.

A comparison of the prediction accuracy of the best predictor presented here with the accuracy of predictors from previous work could be useful to find out whether some important features are missing from the feature set used here. A fair comparison is not easy as the data sets are almost always different. Hirschberg (1993) and Yuan et al. (2005) use a f2b data set that differs from the one we use in that it also includes the the f2b data in the multi-speaker session. The accuracy on a 10-fold cross validation is 82.4% and 83.9% which is lower than the 85.2% accuracy of our best predictor.

Others (Ross and Ostendorf (1996) and Sun (2002) use the same f2b data sets but to predict accented syllables instead of accented words. Chen and Hasegawa-Johnson (2004) and Levow (2008) use the whole BURNC using a "leave one speaker out" training and testing procedure and achieve 82.67 and 84.44% accuracy (when using textual features only). A problem of this "leave one speaker out" procedure in which the predictor is trained on all but one speaker and tested on the left-out speaker is that all the test data (more precisely the textual part of the test data) is included in the training data, so there is a bias that most probably increases the predictor's accuracy.

In general all the prediction accuracies on BURNC reported in previous work are lower than the accuracy of our best predictor.

Concerning the prediction accuracy on SWBDC, the data sets from previous work are different from the corpus used here (the annotation version and the conversations are different). Gregory and Altun (2004) report a 75.67% accuracy while Brenier et al. (2006) achieve a 76.17% when using automatically extracted features. Note that the most predictive features used in Brenier et al. (2006) are the same used in Yuan et al. (2005) to predict accents on BURNC. The accuracy of the best predictor proposed here is 75.84%.

Consistently with the prediction accuracies reported on previous work the accuracy of our predictor on BURNC is definitely better (85.5% vs. 75.84%) than that on SWBDC. Possible causes of such difference are: 1) SWBDC data is less "consistent" than our BURNC data in that BURNC contains the speech of one single speaker while SWBDC contains the speech of tens of speakers; 2) the feature extraction (especially extraction of syntactic features) on BURNC is more accurate that that on SWBDC ; 3) the manual labeling guidelines of the two corpora are slightly different; 4) the main effects on pitch accenting on read speech are different from those on spontaneous speech.

Although we can not draw a definitive conclusion on whether our best predictor

outperforms predictors from previous work we can certainly say that it does not miss any relevant textual information for accent prediction and that a set of features including the best features from previous work and the features proposed here increases accent prediction accuracy.

What emerges when observing the results in the work presented here and in the most recent previous work is that the use of new features and machine learning methods has not drastically increased prediction accuracy. All predictors using automatically extracted features lack of features accounting for a satisfactory description of the discourse context so we might think that this kind of features is what we need to make a leap in accuracy. However studies like Brenier et al. (2006) in which this kind of information was manually annotated indicate that these features lead to a very small increase in accuracy.

This experimental evidence raises the question on whether a natural ceiling on accent prediction accuracy has been reached. The analysis of error at the sentence level of the predictors presented here may provide a partial answer to such question. For example our best predictor sometimes deaccents function or very-frequent words that should instead be accented because the discourse context *focuses* (or contrasts) them. Brenier et al. (2006) reported having the same type of error when predicting on SWBDC and despite using a rich set of features describing Information Structure (e.g., information status,contrast, etc...).

Both accuracy measure and error analysis give an idea of the goodness of a predictor and of its main limits but they do not take into account the variability of pitch accent placement. Several patterns of accenting conveying the same meaning are allowed on the same sentence and a correct evaluation of pitch accent prediction should take that into account. When we compute the predictor accuracy we count as an error the accenting of a word because it is not accented in the test data but that same word in the same text may be accented by another speaker (reading the same text) or by the same speaker if asked to read the text again. As a consequence the gap between our predictor's accuracy and a 100% accuracy is not the real gap[17].

The issue on of variability in pitch accent placements and on the use of accuracy measures that take into account such variability are addressed in the next chapter.

[17]Note that it is accent prediction and not accent detection that is addressed. In the case of accent detection we should aim for a higher accuracy in that variability on pitch accent placement is not an issue anymore. However it may be erroneous to aim for a 100% detection accuracy because of annotation variability (i.e., different annotators can annotate different accent sequences on the same utterances, see section 4.1).

Chapter 4

Optionality in pitch accent placement

This chapter is devoted to the analysis of the phenomenon of variability, and consequent optionality, of pitch accent placement, to the investigation of prediction evaluation metrics that take into account such variability, and to the search of novel methods that take advantage of such variability to improve both segmental and prosodic quality of synthetic speech in unit selection speech synthesis.

Most of the ideas discussed here are actually not specific to variability of pitch accent placement and can be easily applied to other symbolic prosodic events (e.g., prosodic breaks).

In natural speech, alternative prosodic realizations of a given sentence can convey the same meaning. Even when a speaker is required to utter a sentence in a specific standard speech style (that of radio news, for example) she will be free to choose amongst different prosodic patterns all conveying the same meaning. This freedom of choice affects different aspects of prosody, ranging from prosodic phrasing to pitch accent placement. Obviously this freedom has some limits in that not all the acceptable prosodic patterns convey the same meaning. For example the placement of a prosodic break helps to disambiguate the semantic meaning of syntactically ambiguous sentences like "*He ate the cookies on the couch*". Similarly the placement of pitch accents (and of primary accents) can determine the semantic meaning of ambiguous sentences (e.g., *SHE visited me before Sue* = "before Sue visited me" vs. *she visited ME before Sue* = "before she visited Sue", from Rooth (1992)) or can imply different additional (pragmatic) meaning (other than the unambiguous semantic meaning of the sentence) like in *LUKE killed that man* (= Luke, not someone else, killed that man") vs. *Luke killed THAT man* (= "Luke killed that man, he did not kill someone else") (where capitalized words bear the most primary accent in the utterance).

A possible way to study pitch accent variability is that of comparing speech of different speakers reading the same text (and assuming that all speakers convey the same semantic/pragmatic meaning). The first part of this chapter is devoted to a study on variability in accent placements on the section of the BURN corpus (see section 3.4.1) containing speech from different speakers reading the same sentences

Given a sequence of accent values (i.e., a sequence of $\pm$accents) on a utterance not all the accent values are equally subject to variability. Some accent values are strongly constrained by all the main factors affecting accenting (e.g., informativeness, salience) and so are "compulsory" values, while other values are less constrained and so can be seen as "optional". These "optional" accents account for variability in accent placement.

A distinction between "compulsory" and "optional" accents[1] is crucial in the prediction of natural accent patterns. Errors on "compulsory" accent values are much less acceptable than errors on "optional" accent values. To take that into account when evaluating a prediction, we can tune the cost of a prediction error depending on the degree of optionality of the accent value to be predicted. A prediction error on a "compulsory" accent will have a higher cost than an error on an optional accent.

After pointing out the limits of evaluation metrics that consider optionality as a binary variable (i.e., that rigidly distinguish between compulsory and optional accents) we propose an evaluation metric that incorporates an information-theoretic definition of optionality (in which optionality is a continuous-valued variable) and that overcomes some of the limits of previous metrics.

A pitch accent predictor for a TTS voice should be consistent in its predictions, i.e., the accent sequences it generates must be generated as if they were always generated by the same person. A lack of consistency might result in accent sequences that confuse the listener or even cause misunderstandings. Under this perspective the evaluation of pitch accent predictors on multi-speaker data raises a crucial question. Are the accent placements that result to be optional when comparing the speech of different speakers reading the same text also optional in a single speaker's speech? In other words, if a single speaker read the same text several times, would the accent placements resulting optional from a comparison of the different realizations also be optional when comparing the speech of different speakers reading the same text?

Using the information-theoretic formulation of optionality we show some empirical facts suggesting that in general the value of optionality of an accent value (on a given word token) observed when comparing the prosodic patterns of different speakers (henceforth intra-speaker optionality) is very similar to the value of optionality that would be observed on a single speaker (henceforth inter-speaker optionality) if the speaker read the same text several times. That justifies the evaluation of pitch accent predictors on multi-speaker data.

Pitch accent optionality is not necessarily only an obstacle hampering the evaluation of pitch accent predictors but can be turned into something helpful to take advantage of. In section 4.5 we propose a simple method that takes advantage of optionality in order to improve the prosodic realization of a unit selection TTS system that uses pitch accent prediction to model prosody.

[1] We will see later that actually a rigid distinction between "compulsory" and "optional" is not correct, that is why the two terms are in inverted commas.

	f1a	f2b	f3a	m1b	m2b	m3b
the	N	N	N	N	N	N
surveillance	A	A	A	A	A	A
system	A	N	N	N	A	N
is	A	N	N	N	N	N
not	A	A	A	A	A	A
that	A	A	N	A	N	N
sinister	A	A	A	A	A	A

Figure 4.1: *A fragment from the "multi-speaker" section of the BURN corpus. A and N stand for accent (+accent) and no-accent (-accent) respectively. f1a, f2b, f3a, m1b, m2b and m3b are the speaker id's.*

The chapter concludes with a discussion on some issues: the possible extensions of our approach on dealing with optionality to other prosodic events, the working assumption beneath our definition of optionality, its consequent limitations and possible improvements.

4.1 Intra-speaker disagreement

Figure 4.1 shows an excerpt of the "multi-speaker" corpus in which the speech of six speakers reading the same text is prosodically annotated with ToBI conventions (pitch accent ToBI types are collapsed into ±accent). This part of the BURNC was already analysed in Yuan et al. (2005) to investigate the intra-speaker disagreement in pitch accent placement. Here we carry out a more detailed analysis as that is necessary to introduce some of the issues addressed in the next sections.

Figure 4.2 shows the percentages of intra-speaker agreement for each combination of speakers and the agreement mean, with respect to the number of speakers in the combinations, on a text of 1662 words (punctuation marks are not included). The vertical segments range from the lowest to the highest agreement rate for a given number of speakers in the combinations. For example, there are 15 possible combinations of two speakers. Among them the pair with the lowest agreement (79.19%) is f1a-m2b, whereas the highest agreement (85.86%) occurs in pair m1b-m3b. These two percentages may suggest a correlation between degree of agreement and speaker gender, but

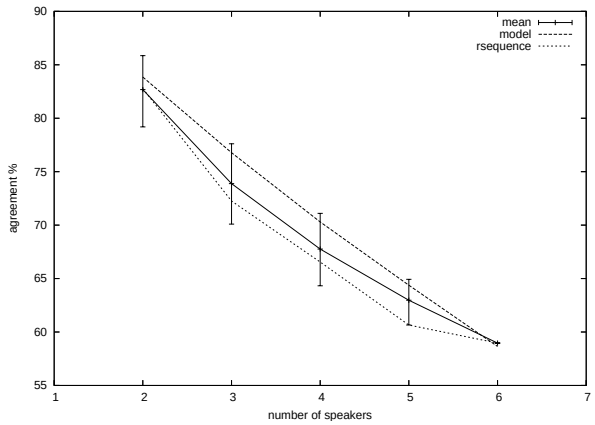

Figure 4.2: *Intra-speaker agreement in pitch accent placement. Each vertical segment represents the agreement rate for a given number of speakers in the combinations. For example the leftmost segment represents the agreement range for all combinations of 2 speakers. The mean line represents the mean disagreement value. The rsequence line shows the disagreement resulting when adding a speakers in the order f1a,f2b,f3a,m1b,m2b,m3b. The model line (the line on the top) is explained in 4.2.1.*

if we look at all the 20 possible triplets of the six speakers we see that the combination with the highest agreement (77.61%) is f2b-m1b-m3b, which consists of one female and two males. We did not carry out any study to investigate the factors that correlate to intra-speaker agreement, but from an informal analysis it seems that speaker profession (is she/he a professional speaker?) is at least as significant as speaker gender. When comparing the agreement among speakers in pitch accent placement we can compute the proportion of agreement that is not due to chance by using the Kappa statistic:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of times speakers agree and $P(E)$ the proportion we would expect them to agree by chance. In our case, assuming that accent and non-accent are equi-probable (the percentage of accented words in this corpus ranges from 45% to 55% depending on the speaker) the κ value for the six speakers (the κ value for "number of speakers = 6" in figure 4.2) is 0.57.

It must be pointed out that part of the computed disagreement in accent placing is due to disagreement between annotators on the presence or absence of a pitch accent. On a study on labeler reliability on a subset of the BURN corpus, Ostendorf

et al. (1995) found an agreement of 91%. The agreement rate was computed on all annotator-pair-words so it is equivalent to the average agreement of all combinations of two annotators. This agreement may "explain" part of speaker-pair-words agreement which is around 83% (see figure 4.2), although we actually do not know how the data to label was distributed between the annotators.

The fact that part of intra-speaker disagreement may be due to annotator disagreement would imply that the optionality of an accent value is not only due to variability in prosody production but also to the variability initrinsic to the annotation process (see discussion at the end of section 2.3).

Since we cannot quantify the effect of annotation variability on the optionality of an accent value we will assume that annotators mainly disagree on accent values on which speakers usually disagree.

4.2 Pitch accent optionality and pitch accent predictors evaluation

Starting from previous studies that first addressed the issue of evaluating the prediction of prosodic symbols taking into account prosodic variability (section 4.2.1) we point out the limits of evaluation metrics based on a binary formulation of optionality and put the basis for the evaluation functions introduced in section 4.2.2.

4.2.1 Previous Work

There is little previous work on the evaluation on multi-speaker data of the prediction of prosodic symbols. Ross and Ostendorf (1996) propose a best match evaluation metric in which for each sentence the predicted sequence of accents is tested on the test sequence that is closest to it. The main limit of this evaluation metric is that at each sentence the evaluation is on one single speaker test accent sequence (the most similar to the predicted sequence) and so it does not capture the relative importance of the different accent values in that it does not distinguish between "compulsory" accent values from "optional" ones.

Ross and Ostendorf (1996) again, Marsi (2004) and Yuan et al. (2005) propose a compulsory/optional evaluation in which prediction errors are only marked when the speakers are unanimous in the placement or absence of a pitch accent (i.e., when the accent value is compulsory according to the test data). If the predicted accent value

has been annotated on at least one speaker then the prediction is always correct, since the symbol is optional.

Yuan et al. (2005) show that when their predictor is evaluated with compulsory/optional evaluation on the multi-speaker section of BURNC its accuracy is very close to 100% and so conclude that the room for improvement on automatic pitch accent prediction is very little.

Such a conclusion may be right but it is drawn assuming that if a pitch accent turns out to be optional when comparing the speech of different speakers then it will be optional in a single speaker's speech as well. As a consequence the optional accent values of a speaker can be swapped with the optional values of (an)other speaker(s) without altering the naturalness of the whole accent pattern.

There are however possible side-effects in this assumption. First, even if all the speakers are unanimous on an accent value that accent value can actually turn out to be optional if more speakers are added in the testing pool. Second, the compulsory/optional evaluation ignores the (possible) interdependence (discussed in previous chapter) between pitch accents, i.e., it ignores that switching optional accent values independently of preceding accent values may result in an unnatural and "distorted" accentuation pattern.

In spite of these (theoretically) possible side-effects, the evaluation function we propose here is based on the same working assumptions of the compulsory/optional evaluation. In the next sections we propose arguments and show empirical facts suggesting that part of these side-effects are not so significant as it may seem at a first glance and can probably be ignored.

Nevertheless, even assuming that such side-effects are irrelevant, the compulsory/optional evaluation metric has still some significant drawbacks. Figure 4.2 shows the steep drop of intra-speaker agreement when the number of speakers increases. As a consequence it is easy to see that the compulsory/optional evaluation metric is strongly dependent on the number of speakers involved.

Figure 4.3 shows this fact by comparing three different predictors: an all-accented predictor (predictor A), an automatic predictor trained on multi-speaker training data (predictor B) (see section 4.3 for more details) and speaker m3b (predictor C)[2]. The three predictors are tested on one of the four parts (which is approximately a quarter) of the multi-speaker BURNC data. Predictor B was trained on the remaining three parts. The accuracy is computed varying the number of speakers involved in the test.

[2]Throughout this chapter punctuation marks are excluded from the evaluations.

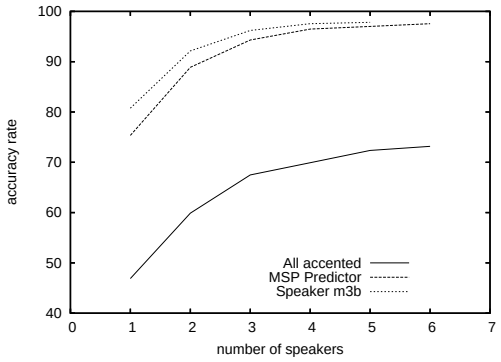

Figure 4.3: *Three predictors tested over combinations of an increasing number of speakers. The sequence of speakers combinations is f1a, f1a-f2b, f1a-f2b-f3a, f1a-f2b-f3a-m1b, f1a-f2b-f3a-m1b-m2b, f1a-f2b-f3a-m1b-m2b-m3b. Predictor A is an all-accented predictor. Predictor B is the predictor MSP described in section 4.3. Predictor C is the speaker m3b.*

The more the speakers in the test data set are, the lower the intra-speaker agreement is and consequently the better the predictor results are.

Consider predictor A, which accents all words. If it is evaluated on six speakers, its accuracy rate is 73%, that means that we could build a predictor that accents the 73% of words, and achieves a 100% accuracy rate. However, since the percentage of pitch accent in read speech ranges from 45% to 55% such a predictor is by far not appropriate to model pitch patterns of real speech.

When looking at intra-speaker disagreement it should be taken into account that the steep decrease is partially due to the simple fact of adding new speakers in the test data, even when the disagreement in each pair of speakers is low. In order to better understand the relationship between number of speakers and intra-speaker disagreement we could try to model it by defining the agreement rate as a function of the number of speakers. By doing that we might have an idea of what the agreement rate is going to be when more than six speakers are compared.

To build the model we could suppose that each word token in the test data has a non-zero probability of being optional, that is of being assigned both accent values and that an accent value is independent of the others. If we assume p being the average probability of the most probable accent value (i.e., N or A) of the word tokens, the agreement percentage can be modelled as:

$$(m1) \quad A(n) = 100[p^n + (1-p)^n]$$

where n is the number of speakers involved. p^n is the probability of having all n speakers agreeing on the most probable accent value on a given word token w_c (e.g.,probability of "YYY", if $n = 3$ and "Y" is the most probable accent value for w_c). $(1-p)^n$ is the probability of having all n speakers agreeing on the less probable accent value on w_c (e.g., probability of "NNN", if $n = 3$ and "Y" is the most probable accent value for w_c). So the sum of the two terms $(p^n + (1-p)^n)$ is the probability of having intra-speaker agreement on w_c. We are supposing that this value is identical for any w_c (although any w_c has its own most probable accent value). In Figure 4.2 the model was plotted setting p to 0.9157. This value was obtained by imposing p^6 (the term $(1-p)^6$ was ignored) as it is equal to the real agreement of six speakers $(0.59)^3$.

Even if the model is certainly a gross model it clearly shows that even for high values of p the agreement percentage rapidly decreases when new speakers are added in the testing data and gives an idea of what might happen if more than six speakers are compared.

The number of speakers is not the only parameter that can affect evaluation. The evaluation function used to evaluate the predictors of figure 4.3 considers correct a pitch event if it is realized by at least one speaker, but, as proposed by Marsi (2004), a "stricter" evaluation function can be chosen by setting a minimum number of speakers agreeing with the prediction. Considering n the number of speakers involved in the test and m (with $m < n$) the minimum acceptable number of speakers that agree with the predictor, than the evaluation function for each word token w_i is[4]:

$$OE(w_i) = \begin{cases} 1 & \textit{if at least m speakers realized} \\ & \textit{the predicted event} \\ 0 & \textit{otherwise} \end{cases} \tag{4.1}$$

Table 4.1 shows the evaluation of the predictors A and B already used in figure 4.3, this time using all the six speakers ($n = 6$) in the test data but varying m.

The high dependency of the evaluation function on m is again explained by the intra-speaker disagreement. When m increases, the number of cases in which the prediction is considered correct independently of its predicted value decreases. For example, if $m = 1$, the prediction is always correct in all the cases where at least one speaker

[3]Note that p was not set to find the best model for the "real" agreement (in terms of maximum likelihood, for example).

[4]Where OE stands for Optional/compulsory Evaluation.

	$m=1$	$m=2$	$m=3$
Predictor A	73.17	65.04	60.16
Predictor B	97.29	92.68	87.80

Table 4.1: *Accuracy rates of two predictors for different values of m ($n=6$). Predictor A is an all-accented predictor. Predictor B is the MSP predictor.*

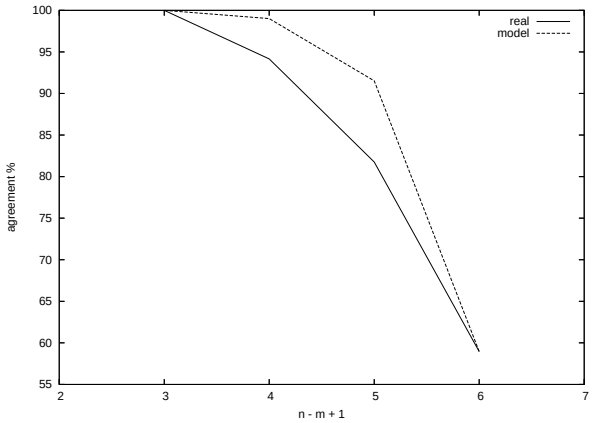

Figure 4.4: *Intra-speaker agreement (real and modelled) for different values of m and fixed n (n=6). $n-m+1$ is the number of speakers (out of n speakers) that agree. Note that the combination of agreeing speakers need not be always the same and can change at each word token. Also, note that for $n-m+1 \leq \frac{n}{2}$ there will always be a combination of $n-m+1$ agreeing speakers.*

disagrees whereas it can be wrong only when all speakers are unanimous and disagree with the prediction. For $m=2$ the prediction is always correct in all the cases where at least two speakers disagree. In general a prediction is wrong only when $n-m+1$ speakers unanimously disagree with it.

Figure 4.4 shows the percentage of accent values that are identical for: all the six speakers (bottom right), at least five out of six speakers and so on, together with a model of agreement (m2) based on the same hypotheses made for (m1). Since the number of combinations of k speakers taken from a set of n speakers is given by $\left(\begin{array}{c} n \\ k \end{array} \right)$, in this case the agreement function is:

$$(m2) \quad A(n,m) = 100 \sum_{k=n-m-1}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) [(1-p)^{n-k} p^k + (1-p)^k p^{n-k}]$$

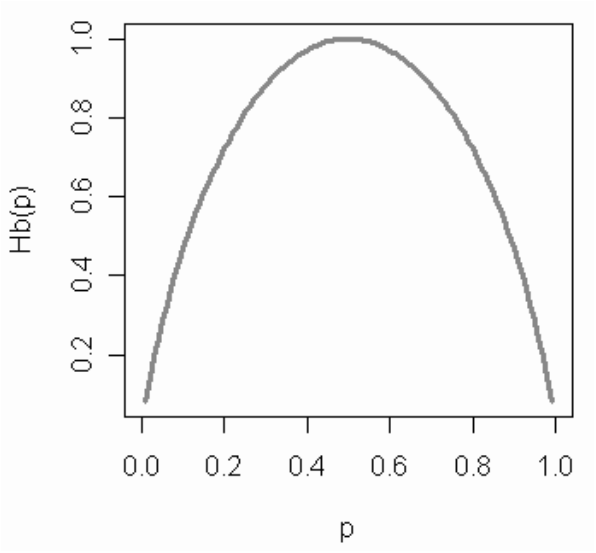

Figure 4.5: *The Binary Entropy function. p is the probability of one of the symbols (with the other symbol having probability $1-p$).*

where $0 < m \leq \frac{n}{2} + 1$,and the p value is again set to 0.9157. When $m \geq \frac{n}{2}$ there will always be $n-m+1$ speakers that agree. $k = n-m+1$ is the number of speakers (out of n speakers) that agree. $(1-p)^{n-k}p^k$ is the probability of having all $k = n-m+1$ speakers agreeing on the most probable accent value on a given word token w_c, and $(1-p)^k p^{n-k}$ is the probability of having all $k = n-m+1$ speakers agreeing on the less probable accent value on w_c. As for model (m1) we are supposing that the sum of these two values is identical for any w_c (although any w_c has its own most probable accent value).

4.2.2 Alternative evaluation functions

The considerations and analysis made above point out the necessity of evaluation metrics that are less dependent on n and m. To be less dependant on n and m an evaluation function must also be less dependent on outliers, which means that when only one speaker (the outlier) disagrees with the majority (on an accent value) the evaluation metric should not drastically change compared to the unanimous case. Such a metric would award predictors able to match the average accent pattern of human speakers.

Here we propose two evaluation functions that satisfy these specifications. Both functions assign a prediction accuracy value at each word w_i of the test data-set that is real-valued (and not discrete as in function OE) and ranges from 0 to 1. The first func-

tion we present assumes that when an accent value has a maximum value of optionality on a word w_i (i.e., when $\frac{n}{2}$ speakers accent a word w_i and the remaining $\frac{n}{2}$ do not) then the prediction accuracy value will be always 1, independently of the prediction, i.e., when an accent value is completely optional it does not matter what the prediction is. In the second evaluation function, the accuracy prediction is linearly dependent on the number of speakers that agree with the prediction and it always increases when the number of speakers that agree with the prediction increases.

To introduce the first evaluation function we reformulate the definition of optionality within the Information Theory framework (Shannon (1948)). First we associate a source emitting pitch accent values to each word token so that each source can emit two symbols, one representing $+$accent and the other $-$accent. The overall number of emissions is equal to the number of speakers and each emission is independent of the others. From Information Theory we know that the entropy of such a source is:

$$H = -p\log(p) - (1-p)\log(1-p) \tag{4.2}$$

where p is the probability that the source emits an accent (i.e., symbol $+$accent) and $(1-p)$ that it does not (i.e., the probability the it emits symbol $-$accent).

The entropy quantifies how much information we need to correctly predict the next symbol that will be emitted by the source. In the case of a source emitting two symbols, if the source has always emitted the same symbol then its entropy will be 0, whereas if the number of emissions of the two symbols is equal then the entropy value will be 1. In all the other cases (and if the number of emissions is higher than 2) the entropy value will be less than 1 and more than 0 (see figure 4.5).

Having associated a word token to a source emitting $\pm$accent, the entropy of the source (H) has the properties that a real-valued measure of the optionality of the accent value for that word should have, that is a maximum value (i.e., 1) when the two accent values are equally probable, a minimum value (i.e., 0) when one of the two accent values has probability 1, and values in-between for all the other cases.

Once optionality has been defined as entropy the next step consists in searching for an evaluation function where the accuracy prediction value depends on entropy. Such specification is satisfied by the following function:

$$tEE(w_i) = 1 - [(1 - P_t(y_i))(1 - H_t(w_i))] \tag{4.3}$$

where $P_t(y_i)$ is the probability of the predicted accent value y_i of occurring at word

w_i and $H_t(w_i)$ $(= -P_t(y_i)\log(P_t(y_i)) - (1-P_t(y_i))\log(1-P_t(y_i)))$ is the optionality of the accent value for word w_i computed on the testing data. EE stands for Entropy Evaluation.

The term $1-P_t(y_i)$ computes the error of the predictor while term $(1-H_t(w_i))$ represents a cost of the error that depends on the optionality value, the more optional the accent value on word w_1 the smaller the cost. For example if word w_1 has been accented by 2 speakers out of 6 in the testing data and the predictor predicts an accent on w_i than the error $1-2/6 = 0.67$ will be reduced by $1-(1-H_t(w_i)) = 1-0.92 = 0.18$. $tEE(w_i)$ is 1 when all the speakers agree with the predictor or when half of the speaker (maximum entropy) agree with the predictor.

The overall tEE is simply the sum of each $tEE(w_i)$ divided by the total number of words.

$tEE(w_i)$ is plotted in figure 4.6 as a function of $P_t(y)$. It satisfies two desiderata: it has a global maximum when the accent values in w_i are totally optional ($P_t(y_i) = 0.5$) and when $P_t(y_i) = 1$. However when $0.5 < P_t(y_i) < 1$ the values of tEE are below 1 because the function is not monotonic and that leads to the paradoxical situation where tEE decreases although the probability of the predicted value, i.e., $P_t(y_i)$, increases.

In order to have a monotonic function we need to modify tEE. A modified version of tEE that preserves the two desiderata mentioned above is the following:

$$EE(w_i) = \begin{cases} tEE(w_i) & \text{if } P_t(y_i) \leq 0.5 \\ 1 & P_t(y_i) > 0.5 \end{cases} \tag{4.4}$$

This is the entropy-based evaluation function we propose.

A potential drawback of EE is that when $P_t(y_i) \geq 0.5$ the prediction accuracy value is always the same independently of $P_t(y_i)$. An alternative evaluation function that avoids that, but at the cost of loosing the important property of having a maximum accuarcy value when optionality is at its maximum (i.e., $P_t(y_i) = 0.5$) is the following:

$$nwEE = 1-(1-P_t(y_i)) = P_t(y_i) \tag{4.5}$$

In $nwEE$ the prediction error $1-P_t(y_i)$ is no more weigthed by the entropy value and $nwEE$ has the disadvantage of not being positively correlated to the optionality the accent value . The function is a strictly increasing function of $P_t(y_i)$, so the higher the proportion of speakers that agree with the prediction, the higher the prediction accuracy.

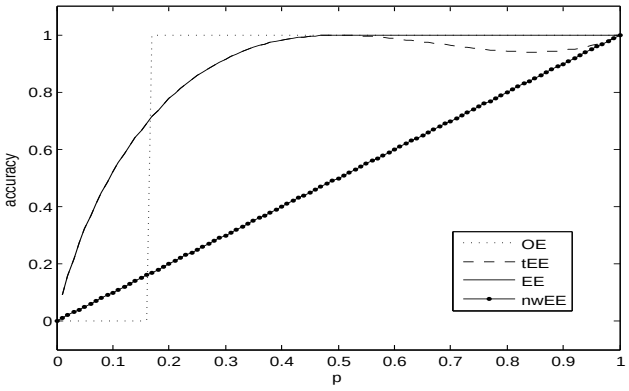

Figure 4.6: *Plots of the evaluation functions OE, tEE, EE, nwEE. p is* $P_t(y_i)$. *OE was plotted setting* $m = 1$.

Henceforth we will consider EE as the evaluation function based on a information-theoretic definition of optionality, and $nwEE$ as the evaluation function that is equivalent to $P_t(y_i)$. Both are plotted in figure 4.6.

One of the practical advantages of EE and $nwEE$ is that there is no m value to be set anymore, while regarding n it is easy to see that they are more stable than OE when n changes. In fact for very large n, it is acceptable to assume a non-zero probability for each token of being assigned both pitch events, especially if we think that annotation errors can be made. Both an all-accented and an all-not-accented predictor would score $OE(w_i) = 1$ (if $m = 1$) per each token though neither of them would match the average accentual pattern of the speakers. When using EE or $nwEE$ both predictors

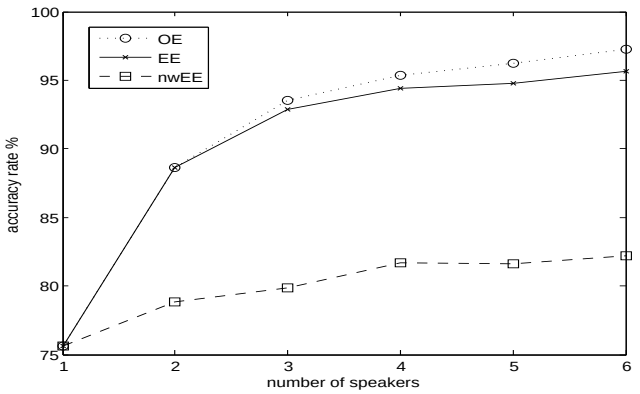

Figure 4.7: *Comparison of OE, EE and nwEE on predictor B. The m value of OE is 1.*

would never reach the 100% accuracy. This is an interesting property of since usually all-accented or all-non-accented predictors can be used as baselines.

In order to provide some empirical evidence of the reduced dependency of EE and $nwEE$ on n, all the evaluations metrics were compared on some predictor. In figure 4.7 predictor B is evaluated on different values of n. When $n > 2$, EE increases slower than OE, while $nwEE$ seems to reach an asymptotic value when $n > 2$. Figure 4.7 shows the results of predictor B evaluated over only one out of 720 possible sequences of speakers. We carried out the same type of comparison using different predictors and different speaker sequences finding always the same trend. An evaluation function that is "stable" with respect to n has also the advantage of guaranteeing that differences in

accuracy between predictors are independent of n.

4.3 Intra-Speaker and Inter-Speaker Optionality

The basic hypothesis justifying the use of metrics that rely on multi-speaker testing data consists in assuming that what turns out to be optional when comparing the speech of different speakers data is also optional within a single speaker's speechIn other words the basic hypothesis is that pitch accent optionality is a speaker independent phenomenon. The fact that, as we have seen in chapters 2 and 3, the main factors affecting accent placement are speaker independent seems to imply that such hypothesis often holds. However this section shows some empirical facts that tend to confirm that[5].

Perhaps the best way to see to which extent such working hypothesis holds would be that of comparing the optionality values (i.e., the entropy values) computed on multi-speaker data and see if they correlate with the optionality values computed on a single speaker data that consists of speech of a single speaker that read several times the same text of the multi-speaker data[6].

Unfortunately when recording the BURN corpus each speaker read the news only once so alternative ways to compute the correlation of intra-speaker optionality and inter-speaker optionality have to be explored.

An alternative approach consists in comparing two different predictors, a predictor trained on single speaker data (henceforth SSP) and a predictor trained on multi-speaker data (MSP, the predictor B seen in previous sections). The motivation of this approach is that the two predictors are representations of the single speaker and of "multiple-speaker" in term of pitch accenting.

SSP is the CART(Wagon)-based predictor described in chapter 3. It is trained on f2b (excluding the f2b part in the multi-speaker data set) and uses three training features: Information Content, Relative Information Content and POS. The MSP uses exactly the same machine learning algorithm and training features used by SSP. It only differs in the training data set which was built by grouping together all the six speakers data of section p, r and t of the multi-speaker data, so the text read by the speakers (1293 words) and the values of the training features were repeated six times (one for

[5]However they are not intended as rigorous proofs of the absolute validity of the hypothesis!

[6]Assuming that in a recording session all text is read by the speaker, at least a few weeks should intervene between a recording session and the following section, as proposed by Chu et al. (2006), for example.

each speakers). As a consequence only the pitch accent values vary. The remaining section, section j (369 words), was held out for testing both predictors.

	e.o. f1a	e.o. f2b	e.o. f3a	e.o. m1b	e.o. m2b	e.o. m3b	e.o. all
SSP	76.15	83.2	82.93	87.26	82.93	84.01	95.21
MSP	75.6	82.11	81.84	87.26	81.3	85.1	95.68

Table 4.2: *Accuracy of a predictor trained on single speaker data (SSP) and of a predictor trained on multi-speaker data (MSP). e.o. stands for "evaluation on". The evaluation on all speakers was carried out using the EE evaluation metric. Punctuation marks are excluded.*

Both SSP and MSP were compared by testing their predictions on each of the six speakers, and on all the six speakers at the same time using the EE evaluation function. Looking at table 4.2, the most evident fact, when comparing the two predictors, is that their performances are very close.

These results can be interpreted looking at a classification tree as a list of prediction rules. We can say, with a certain degree of approximation, that during the MSP training those rules that were sensitive to speakers, that is, appropriate for describing the pitch patterns of some speakers but not for those of the other speakers, were filtered out, so only the rules that assign the non-optional pitch events were successful. If the SSP performances are very close to the MSP ones we can conclude that, at least in our prediction model, the SSP has the same ability of the MSP to distinguish between intra-speaker optional and compulsory pitch events, but this is possible if the inter-speaker optionality "seen" by the SSP during its training phase is very similar to the intra-speaker optionality seen by the MSP.

An alternative way to assess whether the two predictors have "seen" the same optionality (within the prediction model we used) is to compute the correlation of the uncertainties of their predictions. In other words, given a word token, if both predictors are "uncertain" about their prediction for that word token that should mean that the pitch event for that word token is highly optional (the implication "uncertain prediction $\rightarrow$ optional pitch event" is discussed in more detail in section 4.5).

Using again the information-theoretic formulation of entropy, the uncertainty of a

prediction is defined as:

$$\begin{aligned} H_p(w_i) = \quad & -\log(P(y_i|\mathbf{x_i}))P(y_i|\mathbf{x_i}) \\ & -\log(1-P(y_i|\mathbf{x_i}))(1-P(y_i|\mathbf{x_i})) \end{aligned} \tag{4.6}$$

where y_i is the predicted accent value and $\mathbf{x_i}$ is the vector of training features. H_p is computable as the Wagon CART provides, along with the predicted value, the probability of all the possible values (two in our case) of the predicted variable.

The Pearson correlation value of the prediction uncertainties of the two predictors is 0.87. Such a high correlation value (statically significant with $p < 0.01$), suggests again the independence of pitch accent optionality on speakers.

4.4 Human vs. Automatic Prediction Accuracy

The main motivation behind the search for an evaluation metric that incorporates prosodic variability is that of estimating the real gap between natural and automatically-generated accentual patterns.

Because of the requirements on *EE* and *nwEE* of being "stable" metrics (i.e., metrics where the optionality value does not suddenly switch from 0 to 1 when a speaker is added to the testing data) the predictor accuracy computed with *EE* and *nwEE* will never be a 100% accuracy even when the prediction is identical to the accent sequences generated by one of the speakers (unless the intra-speaker agreement is unrealistically 100%). As a consequence the difference between 100% accuracy and the predictor accuracy (computed using *EE* and *nwEE*) is not indicative of the gap between natural and automatically-generated accentual sequences.

To estimate the real gap we used the Bagging-based automatic predictor trained on f2b data (excluding the f2b part in the multi-speaker data set) that in 3.5.1 resulted to be the best predictor on a 10-folds cross validation with 85.2% accuracy (punctuation marks excluded). It uses all the training features described in chapter 3. The *EE* and *nwEE* accuracies were computed on the BURN multi-speaker data with the f2b part held out and were compared with the *EE* and *nwEE* accuracies of the held-out f2b part on the same testing data. The f2b is not actually a prediction but can be fairly considered as it were the prediction of f2b on her own actual generation of accent sequences.

Table 4.3 shows the results of the comparison. The accuracy of the automatic predictor is even slightly higher than the accuracy of the f2b part for any evaluation function (although the difference is very small)

	EE on all-but-f2b	nwEE on all-but-f2b	OE on all-but-f2b	EE on f2b
BAPf2b	96.15%	84.25%	97.2%	84.32%
f2b	96.11%	84.17%	97.15%	100%

Table 4.3: *Human vs. Automatic Prediction Accuracy. BAPf2b is the Best Accent Predictor trained on f2b and described in chapter 3. f2b is the f2b part in the multi-speaker section of BURNC. The accuracy rates in the second, third and fourth column were computed on BURNC multi-speaker data with the f2b part held out. The last accuracy rate is the classic accuracy rate computed on a single-speaker testing data (the f2b part held out from the multi-speaker data).*

These surprising results, although can not be considered as a definitive proof of the full accomplishment of a "perfect" prediction of the accentual patterns can however be regarded as a further proof that the margin for improvement in pitch accent prediction is very little.

From an analysis of error on multispeaker data it emerges that the predictor only fails in few cases, mainly when (in the test data) function words are accented because they convey contrast, when the typical accentuation value of a word is inverted because of some lexical effects, and when content words are deaccented because the concept they convey is redundant. The predictor overaccents (around 60% of words are accented by the predictor while around 50% of words are accented by voice f2b), however that does not seem to be a real problem since the predictor often predicts long sequences of +accent that, although not generated by most of the speakers, are generated by at least one speaker.

4.5 Including Pitch Accent Optionality in Unit Selection Text-to-Speech Synthesis

The ultimate goal of this thesis is that of identifying patterns of prosodic prominence for TTS synthesis. In this section we propose a method that aims to improve the prosodic and segmental speech quality of a unit-selection TTS system by taking advantage of pitch accent optionality.

The core idea is that of associating to each accent (and to each no-accent) place-

ment prediction its supposed degree of optionality, expressed, as we show and motivate later, as the ”uncertainty” of the accent predictor. The working hypothesis is that by incorporating the optionality of the accent value we enlarge the set of prosodically acceptable speech units, and so increase the chances of selecting a good quality sequence of units, both in prosodic and segmental terms. For example, let us suppose that a ”highly optional” accent label has been predicted for a given syllable (of an input sentence to a TTS system). Because of the high optionality of that accent, an unaccented syllable would be probably equally acceptable and so we can allow the unit selection module to select either accent-bearing or no-accent bearing speech units. Doing so we loosen the prosodic constraints without worsening the prosodic model and consequently increase the number of available candidate units.

The advantages of incorporating prosodic variability to improve the quality of unit selection speech synthesis have already been shown in some recent studies (Bulyko and M. Ostendorf (2001), Chu et al. (2006), Campillo and Banga (2006) and Clark and King (2006) among them). For example in Campillo and Banga (2006) different intonation contours are generated via unit selection using prosodic target cost features such as position of the syllable in the intonational group (i.e., prosodic phrase), pitch accent (but without taking into account optionality), etc...[7] The generated contours become then target contours for the standard unit selection phase and the sequence of speech units having the lowest overall (prosodic plus segmental) cost is chosen. Both objective and perceptual tests show the clear benefits of using more than one target intonation contours.

Although exploiting prosodic variability is not a novel idea, the novelty of our approach consists in taking into account the optionality of phonological prosodic events in a way that the automatic prediction of such events is no longer a stand-alone step preceding the unit selection phase, but becomes an integral part of the unit selection process itself.

In order to incorporate pitch accent optionality in a unit-selection TTS system using pitch accent as a feature in its target cost function it could be possible to build an accent optionality predictor trained on the multi-speaker part of the BURN corpus, having the task of correctly predicting the H value per each word token. Then we could associate the predicted optionality value to the accent value predicted by the accent predictor and use it to tune the cost associated to the pitch accent target cost feature.

[7] The different contours are generated by selecting the N-best (with N set to some value) sequences from the Viterbi search algorithm.

Let us consider the standard target cost function for a unit selection speech synthesis system on a target-speech unit pair:

$$T(s_t, u_t) = \sum_{f=1}^{F} w_f(T_f(s_t[f], u_t[f])) \tag{4.7}$$

where $s_t[f]$ and $u_t[f]$ are the values for the feature f of the target unit and the speech unit respectively, T_f is the function evaluating the distance between $s_t[f]$ and $u_t[f]$, and w_f is the weight of the feature f.

Instead of using a standard T_f for the pitch accent feature, that is a T_f that returns 0 when $s_t[f]$ and $u_t[f]$ have the same value and 1 when they have two opposite values, we could introduce the following T_f:

$$T_f = \begin{cases} 0 & \textit{if } s_t[f] \textit{ and } u_t[f] \textit{ are equal} \\ 1 - H(w_i) & \textit{otherwise} \end{cases} \tag{4.8}$$

where $H(w_i)$ is the accent optionality associated to the word containing the target unit s_t.

When $s_t[f]$ and $u_t[f]$ are different and $H(w_i)$ is close to 0, i.e., the accent value is "highly compulsory", then T_f returns a value very similar to the standard T_f, whereas when the accent value is highly optional, i.e., $H(w_i)$ is close to 1, then the cost associated to the pitch accent feature is very low, i.e., it does not really matter if the speech unit comes from an accented syllable or not.

Instead of explicitly using $H(w_i)$ we implicitly incorporated it by weighting the difference between $s_t[f]$ and $u_t[f]$ using the value of uncertainty of the pitch accent prediction on w_i.

Using a probabilistic pitch accent predictor that for each predicted accent value y_i gives the conditional probability $P(y_i|\mathbf{x_i})$[8] the uncertainty of the prediction $H_p(w_i)$ is defined as in equation 4.6.

The main motivation to use $H_p(w_i)$ instead of $H(w_i)$ is that what we actually need to know is how "sure" is the accent predictor about the information it passes to the target cost function. For example, since we deal with not perfect predictors, if we have two separate predictors, one to predict pitch accents and one to predict their optionality, it may happen that the accent predictor assigns a pitch accent to a word with $P(e_i) = 0.55$ and the optionality predictor says the accent is compulsory ($H(w_i) = 0$). In that

[8] A necessary condition on the predictor is that it has to be a predictor in which the probability of y_i does not depend on the preceding accent values y_{i-1}, y_{i-2}, etc... So a CART-based predictor is appropriate while and CRF-based predictor is not.

case what it is important for the unit selection module is that the accent predictor is quite unsure about its prediction and so it does not make a lot of sense forcing the selection module to select accented units, independently from the optionality predictor.

Looking from another perspective, the advantage of using $H_p(w_i)$ is that it is correlates with both the optionality of a pitch accent and the inaccuracy of the accent predictor. In fact our accent predictor does not achieve a 100% accuracy rate because: 1) the set of explanatory features we use is not enough (inaccuracy of the prediction model); 2) prosodic variability occurring within a single speaker's speech. As a consequence $P(y_i|\mathbf{x_i})$ is affected by both factors.

However, the assumption that the H_p of an accent predictor is correlated to pitch accent optionality is not necessarily true and depends on the accent predictor. Making an extreme example, suppose we have an accent predictor trained on a single explanatory feature having two different values: $true$ if the first letter of a word is a d, and $false$ otherwise. In that case it is easy to see that H_p is only due to the inaccuracy of the prediction model. Instead, if the explanatory features are "good enough", i.e., closely correlated to pitch accenting, the optionality observed with respect to those features should be strongly correlated to the real optionality.

In the implementation of this method including H_p in the target cost function, the SSP predictor of section 4.3 was used[9]. Proofs that its H_p is closely correlated to pitch accent optionality are: 1) the features it uses (IC, RIC and POS) have been shown to have a close correlation with pitch accenting; 2) in Badino and Clark (2007) we show that its H_p is by far the best explanatory features of an accent optionality predictor (i.e., a predictor predicting the $H(w_i)$ computed on the BURNC multi-speaker data). The larger H_p is the more optional the accent is.

The efficacy of this method is tested in chapter 6 (experiment 1) where a TTS including H_p is compared on a large scale perceptual test with a TTS having the standard T_f for pitch accents.

[9]We did not use the Bagging-based best predictor described in chapter 3 as the work presented in this chapter was actually carried out before most of the work presented in chapter 3.

4.6 Discussion

4.6.1 Extendability to other prosodic symbolic events

The analysis of variability, the definition of optionality and the use of optionality in the target cost function of a unit selection TTS discussed above can be extended to other categorical prosodic events such as prosodic breaks, ToBI pitch accent types and so on, which being prosodic events are subjected to variability. The definition of optionality we gave is not tied to the concept of pitch accent, it can also be used when the number of symbols is greater than two and it always reaches its maximum when all the symbols are equiprobable.

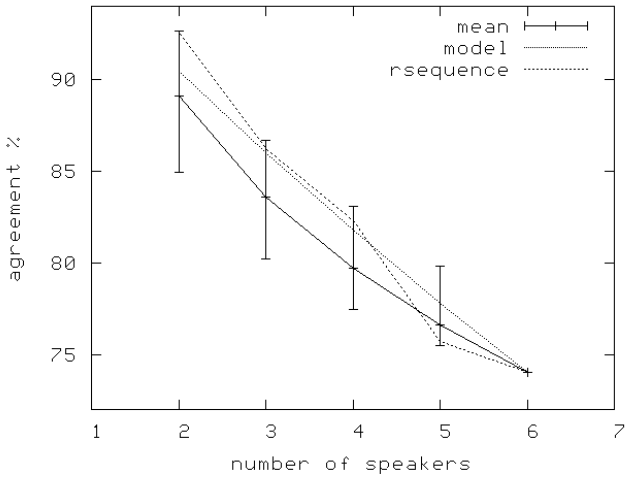

Figure 4.8: *Intra-speaker agreement in phrase breaks placement. The mean line represents the mean disagreement value. The rsequence line shows the disagreement resulting when adding a speakers in order f1a,f2b,f3a,m1b,m2b,m3b. The model line (top line) is computed as the model line of figure 4.2.*

Suppose we deal with phrase breaks instead of pitch accents. If, as it is often done in phrase break prediction, we collapse the five ToBI levels of prosodic breaks to only two different types, i.e., type “break” and type “no-break”, by grouping the phrase breaks of ToBI level 3 and 4 in the “break” type and phrase breaks of ToBI level 2, 1 and 0 in the “no-break” type, the extent of variability observed in BURNC is

comparable with that of pitch accent placements as it is shown in figure 4.8.

The method we propose to exploit pitch accent optionality in unit selection speech synthesis can be applied to prosodic breaks as well. Since the presence of a phrase break in a utterance is usually more perceivable than a pitch accent, and the set of phrase final (and phrase initial) syllables is more sparse than that of accented syllables, our expectation is that improvements obtained using our method (which reduces units sparsity by enlarging the search space) on phrase breaks should be even more significant than when used on pitch accents.

4.6.2 Limits

Throughout this chapter the independence between pitch accents have been assumed. This is a simplification and there may be cases in natural speech where it fails. For example there could be cases where there are two consecutive words whose accenting is optional (when looking at each word independently of the others) but for which a concurrent deaccenting is unacceptable. In such cases we have to remove the assumption of independence in order to allow the configurations AA,AN,NA, but to forbid NN [10].

Even removing the independence assumption is still possible to reuse and reformulate the definition of optionality and the optionality-dependent evaluation metric we proposed by switching from the concept of optionality of a single pitch event to the concept of optionality of a sub-sequence of pitch events delimited by compulsory (or "highly compulsory") pitch events. For example, given the sequences of pitch accents of figure 4.1, instead of looking at the two separate values of optionality of words *system* and *is*, one can look at the bigram *system-is* which is delimited by words accented by compulsory accents, *surveillance* and *not*, and whose alternative accent configurations are AA, NN, AN, which can be seen as 3 out 4 symbols emitted by the bigram-source[11].

The main problem of this reformulation of pitch accent optionality is that by considering sequences of words instead of single words as sources emitting symbols, the number of emitted symbols increases, and so a higher number of speakers in the test data set may be required to avoid data sparsity.

[10] However the number of cases seems to be quite limited since we have seen in the previous chapter that for pitch accent prediction the independence assumption is acceptable as the accuracy of predictors based on this assumption is comparable, if not better, than that of predictors assume accent interdependence.

[11] The fourth symbol NA has probability 0.

Regarding the method proposed to incorporate accent optionality in the target cost function of a unit selection TTS, a reformulation of such method accounting for accents interdependence is not applicable as it would violate the *dynamic programming invariant* (see D. Jurafsky and Martin (2000) for a definition) on which it is based the algorithm that searches for the best sequence (i.e., lowest cost sequence) of candidate units. As a consequence other methods incorporating pitch accent optionality and assuming accent interdependence can be explored in order to see if the independence assumption of our method might somehow worsen the prosodic realization of the synthetic speech.

Chapter 5

Automatic labeling of *contrast*

This chapter concerns the automatic detection of *contrast* based on textual features only. We have defined *contrast* as the relation that links two items that explicitly contrast with (evoke) each other (i.e., "symmetric" *contrast*) or two items where one item contrasts (evokes) the other (i.e., "asymmetric" *contrast*). Because of limits imposed by the data set collected to train the *contrast* tagger described here, the tagger is actually built to identify the instances of *contrast* where the two words linked by *contrast* explicitly contrast with each other and their contrastiveness is prosodically marked.

The goal of identifying *contrast* is motivated by the fact that accents on contrastive words are usually more prominent than "standard" accents (e.g., accents on new information) so that a *contrast* tagger combined with a standard ±accent predictor would allow to go beyond the usual ±accent distinction and consequently would allow to make TTS more expressive and able to generate a context appropriate prosody.

As discussed in section 2.2.3.1, according to the semantic account on *focus* (Bolinger (1961), Rooth (1992) among others) *contrast* is only one of several scenarios of *focus* (i.e., *kontrast*). The annotated corpus we used to train and test the *contrast* tagger is a section of SWBDC whose annotation of Information Structure is described in Calhoun et al. (2005). In the corpus also other scenarios of *kontrast* are annotated: *correction*, *subset*, *adverbial*, *answer* and *other*. A definition of these categories has been given in section 2.5. In the corpus the *contrast* scenario is actually referred to as *contrastive* but throughout the chapter we will use the word *contrast*. In the *contrast(ive)* scenario there are both "symmetric" and "asymmetric" examples and the contrastive word has to be prosodically marked. The identification of all the categories annotated in the corpus, in addition to the identification of *contrast*, might be useful to model prosody in context. However the *contrast* scenario is the closest one to our intuitive idea of contrast, the most studied in terms of prosodic correlates and the most associated to what is generally called *contrastive accent* that is an accent more prominent than the "standard" accent. There are also some practical and "theoretical" reasons to privilege *contrast* over the other categories.

The *correction* category is actually a type of *contrast* (i.e., it is activated by a word/phrase that explicitly contrasts with another), however in the annotated corpus it almost always signals corrections that a speaker makes to her own speech (e.g., "I like a lot, uhm, I like a **little** bit of a lot", where "little" contrasts with "a lot"[1]) and so does not seem very useful for TTS applications (unless the TTS system has to read spontaneous speech, which is not common and requires many other unsolved problems

[1]The reader will note that a few examples from SWBDC shown in this thesis are ungrammatical.

to be solved first).

The *subset* category, which refers to a focused item that is a member of a previously mentioned set (which is evoked by the focused item), seems to us too difficult to be accurately automatically detected since in a lot of cases the "previously mentioned set" is too "abstract" and needs to be inferred from the discourse context while in *contrast* the evoked words are explicitly given.

The *other* category is a "vague" category for which it is very difficult if not impossible to identify which lexical/syntactic/semantic/ factors activate it [2].

Examples of *answer* are too few to build an *answer* tagger, although the category itself, could be very useful. Finally the *adverbial* category simply marks the first function word that follows some adverbs like "only" and "just" so its detection is trivial.

As we have seen in 2.5 this corpus has already been used in Sridhar et al. (2008) to train classifiers able to identify the above *kontrast* categories. Concerning the detection of *contrast* the weak point of the classifier proposed by Sridhar et al. (2008) is that it tries to identify contrastive items without trying to identify the *contrast* relation which says which word contrasts with which word. An approach that ignores the *contrast* relation cannot identify the mechanisms that make two items contrast with each other and so it will never guarantee a satisfactory precision in *contrast* classification. Moreover if we simply know that a word contrasts with some other word without knowing which word it contrasts we might not have enough information to prosodically mark *contrast* in the right way. In other words it might be be possible that it is the *contrast* relation, rather than each single contrastive word, that needs to be prosodically marked. This last aspect will be more clear in the next chapter where the generation of prosodic patterns for contrastive words is discussed.

So far we have referred to the *contrast* identification task as a detection (or labelling, or tagging) task. Following the discussion in section 2.4 on the distinction between detection and prediction of pitch accents the task seems a prediction task rather than a detection task as only textual features are used to predict an event that is also prosodic (as in Calhoun et al. (2005)'s annotation *contrast* is only labelled if the contrastive words are prosodically prominent). However since in TTS applications textual information is the only available information to identify *contrast* we rather look at *contrast* as a relation that is activated by "textual" factors (e.g., semantics, syntax) and that may or may not be prosodically marked according to the speaker's intention.

In the next section we describe the resources and tools used to build a *contrast*

[2] In fact *subset* and *other* are the two categories with the lowest annotation agreement

tagger: the machine learning technique, Support Vector Machine (SVM), on which the tagger is based, and WordNet, the semantic lexicon from which all semantic features have been extracted. Another very important resource for feature extraction is the dependency parser. A description of dependency grammars is given in 3.2.2.

Subsequently we describe the training data and then the *contrast* tagger and the first results achieved. Finally we show various attempts aiming to improve the initial results.

5.1 Resources and Tools

5.1.1 Support Vector Machines

This section does not have the ambition to exhaustively explain what Support Vector Machines (SVMs) are and only aims to highlight the basic concepts, some of which are necessary to introduce the work described in section 5.4. We used SVMs for *contrast* tagging because of their capability of handling a large number of features (which is our case) and of handling structured features like syntactic trees (although we have not exploited this capability) and because they have been largely studied in the field of Active Learning (see section 5.4.4).

SVM (Vapnik (1982),Vapnik (1995)) is a linear discriminative method where the predicted variable $y(\mathbf{x})$ is a linear weighted (by vector $\mathbf{w}$) combination of non linear transformation $\phi_i(\mathbf{x})$ of the feature vector $\mathbf{x}$ (plus a constant):

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{5.1}$$

where $\phi(\mathbf{x})$ is a vector consisting of $\phi_i(\mathbf{x})$ elements and b is a constant.

When using SVM for classification the values of the classes must be numeric. In a binary classification setting the values of the two classes are mapped to +1 and -1 respectively (so in the *contrast* classification +contrast and −contrast are associated to +1 and -1 respectively).

Let us assume for the moment that the training data space is linearly separable, that means that in the transformed feature space it is always possible to find a hyperplane separating all points associated to one class (i.e., all points with target value $t_n = +1$) to those associated to the other class ($t_n = -1$).

In such a case training a SVM consists in finding the values of $\mathbf{w}$ and b such that the hyperplane $y(\mathbf{x}) = 0$ maximizes the *margin*, where the margin is the smallest distance

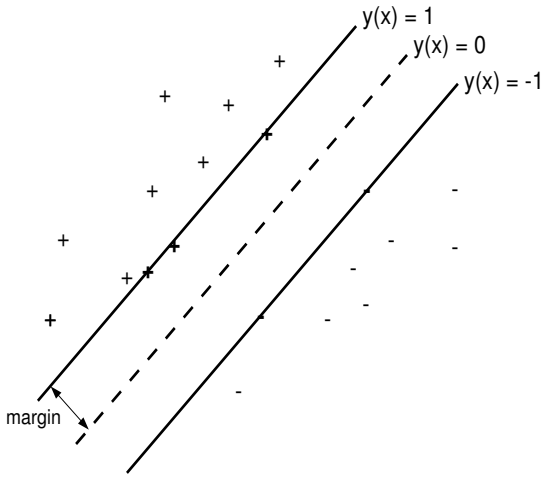

Figure 5.1: *Support Vector Machine - Linearly separable case. Positive and negative examples are linearly separable by the hyperplane* $y = 0$ *(i.e., decision boundary). The* margin *is the maximum distance between the decision boundary and the closest data points.*

between the the hyperplane $y(\mathbf{x}) = 0$ and the closest data points. Figure 5.1 shows an example of an hyperplane maximizing the *margin*. The hyperplane $y(\mathbf{x}) = 0$ is called decision boundary in that all data points $\mathbf{x_n}$ satisfying $y(\mathbf{x_n}) > 0$ are assigned the class value $t_n = +1$, while all data points satisfying $y(\mathbf{x_n}) < 0$ are assigned the class value $t_n = -1$.

Contrary to methods like CART, all features x_i must be numeric. As a consequence symbolic features are mapped into numeric features by mapping each of their symbolic values into a binary feature. For example, considering the POS feature, the value *verb* is mapped into a binary feature which is equal to 1 when the POS is a verb and 0 otherwise.

Intuitively, looking at figure 5.1, given a set of linearly separable data points (with at least 3 data points), there is only one possible hyperplane satisfying the maximum *margin* requirement. The optimal values of $\mathbf{w}$ and b are the values that solve:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{5.2}$$

and satisfy the constraints:

$$t_n(\mathbf{w}^T \phi(\mathbf{x}) + b) \geq 1,\ n = 1,\ldots,N. \tag{5.3}$$

Solving equation 5.2 means finding the value of $\mathbf{w}$ that maximizes the distance between hyperplane $y(\mathbf{x_n}) = +1$ and hyperplane $y(\mathbf{x_n}) = -1$ which is $\frac{2}{\|\mathbf{w}\|}$, while the constraints of equation 5.3 impose that all data points lie on $y(\mathbf{x_n}) = +1$ or $y(\mathbf{x_n}) = -1$ (in the equality case) or on the outside to the *margin*.

The solution to this quadratic programming problem is optimal and is obtained using Lagrange multipliers.

SVM is a kernel method since the classification of the testing data points is carried out using the following equation which is a reformulation of equation 5.1:

$$y(\mathbf{x}) = \sum_{s=1}^{S} a_s t_s k(\mathbf{x}, \mathbf{x}_s) + b \tag{5.4}$$

where $k(\mathbf{x}, \mathbf{x_s})$ is a kernel function defined as the dot product:

$$k(\mathbf{x}, \mathbf{x_s}) = \phi(\mathbf{x})\phi(\mathbf{x_s}) \tag{5.5}$$

The x_s are the support vectors, that is all the training data points lying on the margin hyperplanes $y(\mathbf{x_n}) = +1$ and $y(\mathbf{x_n}) = -1$. The a_s are Lagrange multipliers associated to the support vectors and computed during training.

Popular examples of kernel functions are the linear kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}\mathbf{x}'$ where $\mathbf{x}\mathbf{x}'$ is a dot product, the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}\mathbf{x}' + c)^d$ where c is a positive constant and d is the order of the polynomial, and the Gaussian kernel $k(\mathbf{x}, \mathbf{x}' = exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$.

With respect to many other kernel methods SVM has the important advantage of being a sparse kernel method, which means that when a new data point $\mathbf{x_{new}}$ has to be classified using equation 5.4 the kernel function is only evaluated for all pairs consisting of the new data point and the support vectors (i.e., for all pairs $k(\mathbf{x_{new}}, \mathbf{x_s})$). All the remaining training data points are excluded from the evaluation.

When the transformed training data is not linearly separable as shown in figure 5.2, slack variables are introduced. A slack variable ξ_n is a positive variable defined as $\xi_n = |t_n - y(x_n)|$. Points lying on the correct side (i.e., points where $t_n y_n > 0$) have $0 \leq \xi_n \leq 1$, while points lying on the wrong side have $\xi_n > 1$.

Equation 5.2 is reformulated as follows:

$$\arg\min_{\mathbf{w},b} C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2 \tag{5.6}$$

subject to the new constraints:

$$t_n y_n \geq 1 - \xi_n,\ n = 1, \ldots, N. \tag{5.7}$$

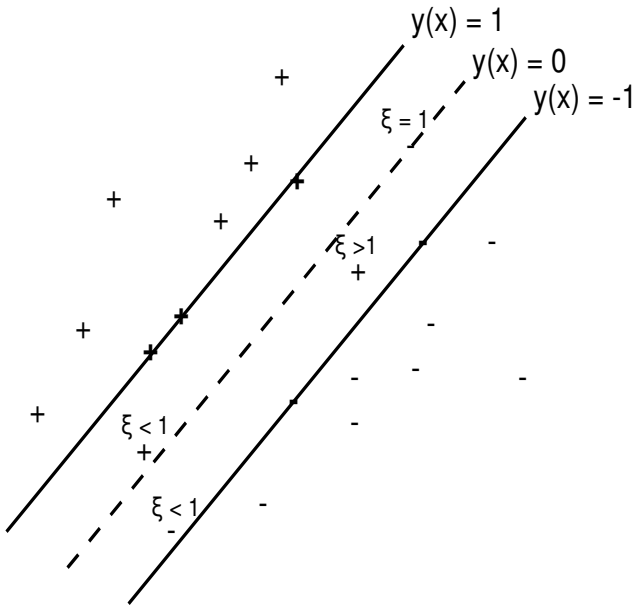

Figure 5.2: *Support Vector Machine - Non separable case. Positive and negative examples are not linearly separable by any linear hyperplane. The slack variable* ξ *associated to each data point is* > 1 *if the data point is misclassified, and* $0 < \xi < 1$ *if the data point is correctly classified and lies in the* margin*.*

The parameter C in equation 5.6 controls the trade-off between generalization (i.e., accuracy on the testing data) and capability (i.e., accuracy on the training data). A high value of C forces the SVM to have a high accuracy on the training data and, as a consequence, could cause overfitting. If $C \rightarrow \infty$ the position of the decision boundary could be heavily affected by just few outliers in the data set, even in a data set of thousand of data points. On the other hand a near-to-zero value of C could cause lack of learning.

5.1.2 The WordNet semantic lexicon

WordNet (Fellbaum (1998)) is an English lexical database. It consists of three separate lexicons: one for nouns, one for verbs and one for adverbs and adjectives. For each lexicon entry (i.e., an orthographic word) all its senses are listed. Each sense is expressed by a group of synonyms (synset), a gloss and some example uses. For example the noun *flash* has the following WordNet entry (ten senses):

1. flash - (a sudden intense burst of radiant energy)
2. flash - (a momentary brightness)
3. flash, flashing - (a short vivid experience) ”a flash of emotion swept over him”; ”the flashings of pain were a warning”
4. flash - (a sudden brilliant understanding) ”he had a flash of intuition”
5. blink of an eye, flash, heartbeat, instant, jiffy, split second, trice, twinkling, wink, New York minute - (a very short time (as the time it takes the eye to blink or the heart to beat)) ”if I had the chance I’d do it in a flash”
6. ostentation, fanfare, flash - (a gaudy outward display)
7. flare, flash - (a burst of light used to communicate or illuminate)
8. news bulletin, newsflash, flash, newsbreak - (a short news announcement concerning some on-going news story)
9. flash - (a bright patch of color used for decoration or identification) ”red flashes adorned the airplane”; ”a flash sewn on his sleeve indicated the unit he belonged to”
10. flash, photoflash, flash lamp, flashgun, flashbulb, flash bulb - (a lamp for providing momentary light to take a photograph)

WordNet provides semantic relations for each sense, that is for each synset. The three sub-lexicons have different sets of *is-a* relations as shown in tables 5.1, 5.2 and 5.3 (from D. Jurafsky and Martin (2000)).

Relation	Definition	Example
Hypernym	From concepts to superordinates	$breakfast \rightarrow meal$
Hyponym	From concepts to subtypes	$meal \rightarrow lunch$
Has-Member	to their members	$faculty \rightarrow professor$
Member-Of	From members to their groups	$copilot \rightarrow crew$
Has-Part	From wholes to parts	$table \rightarrow leg$
Part-Of	From parts to wholes	$course \rightarrow meal$
Antonym	Opposites	$leader \rightarrow follower$

Table 5.1: *Noun relations in WordNet*

WordNet can be seen as a hierarchy of concepts (i.e., synsets) regulated by these *is-a* relations.

Several tools have been provided to browse WordNet. It can be browsed using a Web browser or by using packages or libraries such as the WordNet::QueryData Perl package (Pedersen et al. (2004)).

5.2 Experiment 1 - Semi-automatic *contrast* labeling

This section describes how the training data for *contrast* classification was collected and describes a first *contrast* classifier that uses a combination of automatically and manually extracted features.

Relation	Definition	Example
Hypernym	From events to superordinate events	$fly \rightarrow travel$
Hyponym	From events to their subtypes	$walk \rightarrow stroll$
Entails	From events to the events they entail	$snore \rightarrow sleep$
Antonym	Opposites	$increase \leftrightarrow decrease$

Table 5.2: *Verb relations in WordNet*

Relation	Definition	Example
Antonym	Opposites	*heavy* $\leftrightarrow$ *light*

Table 5.3: *Adjective and adverb relations in WordNet*

5.2.1 Data preparation

This section and the next two sections (5.2.2 and 5.2.3) describe how the training data were collected and what kinds of restrictions were imposed on the examples of *contrast*. All the syntactic information we used to train the tagger comes from the PennTreebank (Marcus et al. (1993)) manual annotation of the Switchboard corpus. This choice was made in order to explore the potential of the tagger as much as possible independently of the errors of the modules it receives information from.

5.2.2 Data collection

Before merging the syntactic and the information structure annotations the syntactic constituent format was converted into dependency trees using the Penn2Malt converter (Nivre (2006)). Since the PennTreebank constituent annotation for Switchboard uses slightly different (and not yet standardly held) conventions from those presupposed by the Penn2Malt converter we had to support the converter with some additional scripts. However, because of problems encountered in the conversion process we had to remove 54 (out of 146) dialogues. The use of syntactic dependencies that were derived from manually annotated constituents makes this first experiment semi-automatic, i.e., not fully automatic[3]. In a second experiment (section 5.3) syntactic dependencies provided by a dependency parser are used instead of gold standard dependencies in order to make the *contrast* tagger entirely automatic.

For each remaining dialogue all the word senses (according to the WordNet senses set) were disambiguated using the WordNet::SenseRelate Perl module (Patwardhan et al. (2005)).

5.2.3 Data pruning

Not all the sentences of the 92 dialogues and not all the examples of *contrast* were used to train and evaluate the tagger. First, for reasons of computational efficiency we

[3]In the fully automatic setting all features are automatically extracted.

decided to only consider *contrast* relations that occurred within a single dependency tree (i.e., within a single sentence, whose boundaries were given by the PennTreebank constituent annotation).

Then, we removed all the sentences that did not contain *contrast* within a single dependency tree. As a consequence all the positive and negative examples that were subsequently generated were only extracted from sentences containing at least one *contrast*.

Subsequently, we decided to consider *contrast* relations linking single words only, so sentences only containing *contrast* linking phrases of more than one word were removed. This decision was dictated by the need of approaching the problem of the identification of contrastive items starting from its simplest instance, i.e., the case where the two contrastive items are single words. We also decided, in order to make the tagger task a bit simpler, to only look at *contrast* that links words having the same broad POS where the broad POS are: noun, verb, adjective, adverb, pronoun, cardinal number, other. This pruning regarded a very small number of *contrastive* word pairs.

After these pruning steps all examples of *contrast* were examples of the "symmetric scenario" of contrast in which the two words explicitly contrast with each other while there were no cases of "asymmetric" *contrast* in which only one word contrasts (backward) with the other (and the other way around is not true). Thus, since our tagger is only trained on the symmetric scenario, one might assume that the tagger does not scale up to both scenarios. However, at least when only looking at *contrast* in text (i.e., *contrast* is only activated by textual factors), the main difference between the two scenarios only resides in the fact that one scenario (the symmetric scenario) almost always occurs within the utterance(s) of the same speaker while the other scenario occurs when one speaker intends to contrast something that has been said by the other speaker[4]. However the main factors activating the two scenarios (e.g., syntactic parallelism, antonymy, etc ...) are the same so a tagger that accurately identifies the symmetric examples should be as much good in identifying the asymmetric examples.

Since the *contrast* tagger has to rely on textual features only and does not look at the discourse context outside the sentence containing *contrast*, we additionally removed: 1) all *contrast* relations that we could not identify by simply looking at text, and that had been labeled only because they were prosodically signaled; 2) all *contrasts* acti-

[4]In order to be symmetric a *contrast* must contain two words that explicitly contrast with each other so that when a speaker utters the first occurring contrastive words already must already know which the second occurring contrastive word is. That is much more probable when it is the speaker herself that utters the second occurring contrastive word

vated by discourse items outside the sentence. In this last pruning step some decisions were hard to make, since *contrast* resulted in a combination of prosodic, syntactic, semantic and pragmatic clues. When we were not sure about keeping or removing the sentence containing the problematic *contrast* we did not remove it.

Note that we did not remove cases where *contrast* was neither syntactically or semantically determined but only determined by pragmatic factors that go beyond the linguistic content of the discourse like in the following example:

(1) as a **westerner** in **India** ... I was often surprised ...

The final data used to generate positive and negative examples of contrastive word pairs consists of 254 sentences containing at least one *contrast*, i.e., positive example, that was not pruned out.

Note that, according with the selection and pruning rules we applied, some of the 254 sentences may contain examples of contrastive word pairs that are identifiable by looking at text only but were not labelled as contrastive since they were not prosodically signaled and so are labeled as negative examples. For example in the following sentence:

(2) My impression of it is that it has **doubled** in the last ten years and **tripled** in the last twenty.

"doubled- tripled" are a positive example of contrastive pair in the training data but also "ten" and "twenty" can be regarded as a contrastive pair depending on the discourse context. Nevertheless "ten" and "twenty" are labeled as negative example in the training data[5].

5.2.4 Examples extraction

For each sentence both positive and negative examples were extracted as shown in Fig.5.3. All word pairs having the same broad POS were extracted as and then assigned a +1 if the two words were linked by *contrast* or a -1 otherwise. An example consists of its positive or negative value and a sequence of training feature values.

The fact that the computation of some features requires a considerable computational effort (but still reasonable for real time applications) and sentences can be 80

[5]No modification on the manual labeling was carried out to avoid potential "subjective" biases in the training data

W1	W2	Example value
we	they	+1
seemed	be	-1
seemed	doing	-1
seemed	cooking	-1
	...	
cooking	enjoying	+1
	...	

Figure 5.3: *Example values generation for* contrast *labelling. Example values generation from the sentence:* ***We*** *seemed to be unfairly doing all the* ***cooking they*** *were doing all the* ***enjoying***. *The example value is defined only when the two words (W1,W2) have the same gross POS. The example value is positive (+1) if W1 and W2 are linked by a* contrast *and negative (-1) otherwise. The figure only shows some defined example values.*

words long or more explains the decision of limiting the *contrast* relations to those occurring within a single sentence.

The overall set of examples consists of 8602 examples: 275 positives and 8327 negatives.

5.2.5 Feature Extraction

The features extracted can be grouped into three categories: lexical features, syntactic features and semantic features. For sake of simplicity hereafter we will refer to the two words of each word pair as W1 and W2, where W1 precedes W2 in the sentence.

Several features describe shared properties/items of the clauses containing W1 and W2, when W1 and W2 belong to two different clauses. For such reason each sentence is segmented into clauses where each clause is a part of a sentence that refers to "verb-dominated" sub-trees. "Verb-dominated" sub-trees are parts of the dependency tree that have a verb (either finite or non-finite) as a root. For example, in the sentence:

(3) So well... **you** take this subject much more personally than **I** do, I suppose.

"So well... you take this subject much more personally than", "I do" and "I suppose" are all clauses dominated by the verbs "take", "do" and "suppose" respectively.

5.2.5.1 Lexical features

Examples of lexical features are: *all CAP words between W1 and W2, first CAP word preceding W1, first CAP word preceding W2, first two CAP words preceding W1, first two CAP words preceding W2*. CAP words are Conjunctions, Adverbs and Prepositions.

These features are intended to capture single words or bigrams that activate *contrast*, like, for example, the bigram "rather than" in the sentence:

(4) So she's going to **sell** it rather than **trade** it in.

A feature to measure the degree of textual parallelism between the two clauses containing W1 and W2 (when W1 and W2 belonged to two different clauses) was also used since textual parallelism can be a clue of *contrast* like in the following example:

(5) ... let's do **this** way, let's do **that** way ...

The parallelism score (normalized) is computed using the Wagner & Fischer edit distance to compare strings of text as proposed by Guegan and Hernandez (2006).

5.2.5.2 Syntactic features

All syntactic features are POS, syntactic dependencies and features derived from both of them. Examples of features derived from POS are the features indicating if W1 is the only word in the sentence having the same broad POS of W2, and the feature indicating if W1 is the closest leftward word preceding W2 and having the same broad POS.

The use of syntactic dependencies (and information related to them), is motivated by the need of identifying syntactic patterns of contrastiveness that can not be identified using POS and lexical features alone. For example knowing that W1 and W2 have the same type of dependency with their heads as in example (3) (both "you" and the first "I" have a "subject of" dependency with "take" and "do" respectively) or that their heads refer to the same item as in example (6), seems to be a necessary (although often not sufficient) information to identify *contrast*.

(6) and, you know, even the **public** schools are behind the **parochial** schools.

In order to improve the detection of parallelism for two words belonging to two different clauses, the feature set contains features indicating if the two clauses have subjects

referring to the same item. The same type of feature is used for syntactic objects, dominant verbs and predicates.

The Wagner & Fischer edit distance based parallelism score mentioned above is also also used to compute the similarity of the sequences of POS (instead of sequences of words as in section 5.2.5.1) of the two clauses containing W1 and W2.

5.2.5.3 Semantic features

Semantic features are often a necessary information to detect *contrast*, since contrastive words are usually semantically similar as in:

(7) and you see **women** going off to work as well as **men**.

and/or are often linked by a particular semantic relation as in

(8) Every time we'd get a real **good** player they'd treat him **bad**.

where "good" and "bad" are antonyms.

On the other hand if a word is the hypernym of another word (e.g., "cat" and "animal") a *contrast* between the two words is very unlikely.

The semantic features consist of a semantic similarity score and a set of semantic relations all computed using the WordNet::QueryData and WordNet::Similarity (Pedersen et al. (2004)) Perl modules.

The semantic relations indicate if W1 and W2 are (or are not): hypernyms, antonyms, entails, member-of, part-of, sisters (i.e., two words having the same hypernym).

The semantic similarity score is the Lin's semantic similarity measure (Lin (1998b)) applied to WordNet. The semantic similarity of two senses is given by the semantic concepts (i.e., senses) the two senses share. The Lin's semantic measure, like other measures of semantic similarity on taxonomies like WordNet, searches for the Least Common Subsumer (LCS) which is the most specific ancestor of the two senses, i.e., the most specific sense that subsumes them. The Lin's similarity measure is given by the Information Content of Concept(ICC) (see 3.2.1) of the LCS scaled by the sum of the Information Content of Concept of the two senses. So the more specific the LCS the highest the score. The similarity of two senses also depends on how close in the taxonomy the two senses are with respect to the LCS, which is why in the Lin's measure the ICC of the LCS is scaled by the sum of the two senses.

Since WordNet relations and similarity measures relate to word senses, they were computed in two different way:

1. on the senses (one per word) provided by the word sense disambiguator (see section 5.2.2);

2. on the first 3 most frequent senses (or less if the word had less than 3 senses) of each word, so a maximum of 9 sense pairs are compared.

In the latter case, the selected similarity score is the highest of nine scores, while the selected semantic relation is (if any relation is found) the one linking the most frequent senses.

5.2.5.4 Full feature set

The following is the complete list of all training features:

- is W2 the leftward closest word to W1 with POS(W2) = POS(W1)?
- is W2 the only word with POS(W2) = POS(W1)?
- is W1(W2) a generic word (e.g., *something*,*elsewhere*,...)?
- CAP word between W1 and W2
- first CAP word preceding W1
- first CAP word preceding W2
- first two CAP words preceding W1
- first two CAP words preceding W2
- prefixW1 where length(prefixW1) $>$ 1 AND W1 = "prefixW1+W2" (only in exeperiment 2, sectio 5.3)
- prefixW2 where length(prefixW2) $>$ 1 AND W2 = "prefixW2+W1" (only in exeperiment 2)
- suffixW1 where length(suffixW1) $>$ 1 AND W1 = "W2+suffixW1" (only in exeperiment 2)
- suffixW2 where length(suffixW2) $>$ 1 AND W2 = "W1+suffixW2" (only in exeperiment 2)
- is W1 = W2?

- LinSimilarity(S1,S2) where S1 and S2 are respectively the "disambiguated" senses of W1 and W2
- Max LinSimilarity(W1,W2)
- SemanticRelation(S1,S2)
- SemanticRelation(W1,W2)
- GrossPOS(W1) (which is identical to GrossPOS(W2))
- POS(W1)
- POS(W2)
- POS(W1)-POS(W2) (this is a POS bigram)
- POS($W1_{-1}$)-POS(W1) ($W1_{-1}$ is the the word preceding POS(W1))
- POS($W2_{-1}$)-POS(W2)
- POS($W1_{+1}$)-POS(W1)
- POS($W2_{+1}$)-POS(W2)
- POS($W1_{-1}$)-POS(W1)-POS($W1_{+1}$)
- POS($W2_{-1}$)-POS(W2)-POS($W2_{+1}$)
- DSD(W1) (i.e., name of the syntactic dependency in which W1 is the dependent)
- DSD(W2)
- DSD(W1)-DSD(W2)
- is DSD(W1) = DSD(W2)?
- Path in the dependency tree between W1 and W2
- Do W1(W2) dominate W2(W1) in the dependency tree?
- is W2(W1) the closest ancestor to W1(W2) in the dependency tree with GrossPOS(W2) = GrossPOS(W1)?
- are W1 and W2 "or disjointed" (e.g., *a CAR or a good BIKE is what I need*)?

- is W1(W2) negated?
- W1 = W2 AND W1 OR W2 is negated?
- do W1 and W2 belong to two different clauses that have the same subject?
- do W1 and W2 belong to two different clauses that have the same main verb?
- do W1 and W2 belong to two different clauses that have the same object?
- do W1 and W2 belong to two different clauses that have the same predicate?
- is DSD(W1) = DSD(W2) AND Parent(W1) and Parent(W2) two different tokens but same word type?
- name of DSD(W1) when DSD(W1) = DSD(W2) AND Parent(W1) and Parent(W2) two different tokens but same word type
- name of syntactic dependency when W1 and W2 are parent of the same children to which are linked with the same dependency
- conjunctions (if any) that link the two clauses containing W1 and W2
- lexical parallelism score of the two clauses containing W1 and W2
- syntactic parallelism score of the two clauses containing W1 and W2
- are the two clauses containing W1 and W2 "adjacent", i.e., does the clause containing W2(W1) immediately follow and is not dominated by the clause containing W1(W2)?

5.2.6 Evaluation

As mentioned above the *contrast* tagger is a SVM based predictor. We used the SVM-light implementation (Joachims (1999a)) which allows to use different types of kernel: linear, polynomial, radial basis function (of which the Gaussian kernel is a special case), sigmoid tanh.

The tagger was evaluated using the leave-one-out estimation of accuracy to make up for the small number of examples of *contrast*.

The polynomial kernel turned out to be the most effective one. Table 5.4 shows the values of accuracy, and precision and recall on the positive examples, for different

orders of the polynomial kernel. The quadratic polynomial gave the best result. A possible explanation for the supremacy of the quadratic polynomial is that the non-linear transformation of the data allows to identify "combinations" of training features that are correlate with *contrast*, whereas the linear polynomial is not able to identify such combinations. However polynomials with higher order seem to overfit the data. This finding motivates the experiment described in sections 5.4.1 and 5.4.2.

The very unbalanced numbers of positive and negative examples induced us to try different values of the SVM-light training parameter j which is the ratio between the cost on false negatives and the cost on false positives. The cost on false negatives (i.e., the cost on examples that actually are positive) and the cost on false positives (i.e., the cost on examples that actually are negative) are obtained by splitting the C parameter of equation 5.6 into two components, one that penalizes false negatives and one that penalizes false positives (see Morik et al. (1999)). The ratio $j = 2$ gave the best results. Trying values of j higher than 1 was also motivated by the presence in the data set of examples that can be regarded as "false negatives".

"False negatives" in the data were mainly due to two reasons. First in a few sentences containing *contrast* between two words also *contrast* between phrases occurred, but the training examples extracted from them were labeled as negatives. For example in the following example:

(9) ... **I** 'm *Debbie More*, you know, may I ask you *who* **you** are and ..

"Debbie More" actually contrasts with the second "you" but "Debbie More" consists of more than one word (although it is one single entity). As a consequence in the training data set we have Debbie - who $\rightarrow$ -1 and More - who $\rightarrow$ -1. Limiting the sentences to those only containing *contrast* between two words would have been preferable but such a constraint would have drastically reduced the number of positive examples.

Second, prosodically unmarked *contrast* was not manually annotated as it is shown in example (2). With respect to the task which consists in finding prosodically prominent contrastive word pairs, non-prominent contrastive word pairs are actually "true negatives" but since the prosodic dimension is hidden to the tagger as it only uses textual features, the non-prominent contrastive word pairs are "false negatives" in the textual dimension.

The analysis of error of the *contrast* tagger is postponed to the next session where a fully automatic tagger is evaluated.

d	j	Accuracy	Precision	Recall
Baseline		96.80%	0%	0%
1	2	97.02%	70.21%	12.00%
2	1	96.88%	65.22%	5.54%
2	2	**97.19%**	**76.19%**	17.45%
2	3	97.17%	65.59%	**22.18%**
3	2	97.00%	68.09%	11.64%

Table 5.4: *Leave-one-out evaluation of the semi-automatic contrast tagger. d is the order of the polynomial kernel. j is the ratio between the cost on false negatives and the cost on false positives. Precision and recall are relative to positive examples. The baseline is a majority baseline always assigning -1.*

5.3 Experiment 2 - Automatic *contrast* labeling

In this second experiment, automatically extracted (by the dependency parser MaltParser (Nivre et al. (2007))) syntactic dependencies were used, instead of gold standard dependencies. With respect to the previous experiment some sentences containing contrastive word pairs were removed because: 1) MaltParser split the sentences in two or more sentences so that the contrastive words did not belong to the same sentence anymore; 2) we removed a couple of sentences where we actually did not identify any *contrast* (and that we mistakenly kept in the training data set of experiment 1). All sentences used for experiment 2 are shown in appendix A. In this new data set, there are 246 positive examples and 7405 negative examples.

Concerning training features, apart from switching from manually to automatically extracted syntactic dependencies, small changes were made, mainly consisting of a couple of small bug fixings, and the introduction of morphological features that say if one of the two words in the pair is contained by the other one (e.g., formal vs. *in*formal) and, in case that is true, indicate the morpheme that differentiates the two words (e.g., *in*). No semantic disambiguation was carried out.

Table 5.5 shows results using second order and third order polynomial kernels. Other machine learning methods were used, but they performed either slightly worse (C.5 classification tree, and RIPPER from Weka (Hall et al. (2009)) or much worse (Bayesian Logistic regression from Weka). Note that results are improved with respect to semi-automatic tagger. The second-order polynomial is still the better kernel.

d	j	Accuracy	Precision	Recall
Baseline		96.78%	0%	0%
1	2	97.05%	64.71%	17.89%
2	2	**97.28%**	**76.39%**	**22.36%**
3	2	97.24%	75.36%	21.14%

Table 5.5: *Leave-one-out evaluation of the fully automatic contrast tagger. Best results are obtained with the 2nd-order polynomial kernel*

Evaluation	Precision	Recall
Leave-one-out	76.39	22.36%
Leave-one(sentence)-out	74.08	24.4%

Table 5.6: *Leave-one-out vs. leave-one(sentence)-out evaluation.*

A fairer evaluation of the tagger is a leave-one(sentence)-out evaluation in which the testing data set consists of all the examples from one held-out sentence instead of consisting of just one example. It is fairer than the standard leave-one-out in that it tends to have the same ratio of positive and negative examples in both testing and training data. Table 5.6 shows the tagger accuracy computed with the two types of evaluation.

Unfortunately there are no tools, when using SVM, to see what the most significant features are. We could compute the correlation of each feature with the target value (i.e., $\pm$*contrast*) but that would not be very informative. In fact some features could be highly correlated to the target value but be redundant once combined with all the other features. So we used the RIPPER classifier which performs slightly worse than SVM but generates human-readable rules. RIPPER extracts IF-THEN rules in the following way: it first searches for the atomic rule, i.e., a rule looking at a single feature value in the *IF* part (e.g., "IF POS = verb THEN *contrast*), that has the best score, where the score is computed using some metric which is a function of the number of times the rule is correct and the number of times it is wrong. Then the rule is grown if its score increases by adding a new element in the *IF* part. If the rule can not be grown anymore, all the examples covered by that rule are removed from the training set and a new rule is searched for in the reduced training set. The final set of rules is pruned to

avoid overfitting.

The rules extracted by RIPPER on our training set, together with the number of times the rule is correct and the number of times it is wrong, are shown below:

1. if (W1,W2) are antonyms → *contrast* (40, 9)

2. if (W1,W2) have a common dependent to which they are related with a "Noun-Modifier" relation AND W2 is the closest word on the left side of W1 having its same POS AND the Lin similarity in terms of POS of the two clauses containing W1 and W2 respectively is > 0.5 → *contrast* (11, 4)

3. if common POS is CardinalNumber AND W2 is not a depend-ant in the dependency PrepositionModifier AND (W1,W2) are not "OR disjointed" → *contrast* (11, 4)

4. if common POS is Noun AND (W1,W2) are depend-ant of the same type of dependency AND have the same head in the PrepositionModifier dependency *contrast* (9, 2)

5. if W2 is a subject (i.e., the depend-ant in the Subject dependency) AND "than"is the first closest CAP word on the left side of W2 AND W2 is the closest word on the left side of W1 having its same POS *contrast* (9, 1)

6. otherwise *no-contrast* (7654, 181)

Knowing if two words are antonyms certainly helps. But when the contrastive words are not antonyms then combinations of syntactic and lexical features are necessary to improve accuracy.

5.3.1 Analysis of Error

The main problem of the *contrast* tagger is its poor recall which is due to different causes. One of these causes is that the feature set lacks of pragmatic knowledge which is sometimes necessary to identify *contrast*. Consider the following two examples:

(a) As a **westerner** in **India**, I was often surprised, and felt my sense of privacy there was quite invaded.

(b) They've quoted statistics that my **throat** just about fell into my **toes**.

where clearly *contrast* is not identifiable if one does not know that the Western and the Indian culture are quite different (a) or does not know that in the human body the throat and the toes are at its two extremes (b).

Contrast can also be activated by the semantic meaning of the sentence containing the two contrastive words rather than by more "superficial" syntactic parallelism or phrases like "rather than", "instead of", etc... In the example:

(c) The parents might have hostilities towards them , like **you**'re judging **us** ,

contrast is activated by the clause preceding the two contrastive words and also by the verb "to judge" which in some contexts can imply confrontation between the subject and the object.

As for the case of "rather than", "more than", and so on, single or bigrams of words are sufficient to activate *contrast* like in the sentence:

(d) Their attitude and philosophy was just completely , opposite from **mine**...

where "opposite from" is a clear clue of *contrast*.

The very limited number of positive examples does not allow the tagger to identify these clues as they only occur once in the training data. In general, the small set of positive examples often includes just one example of a scenario of *contrast*, too few to identify the relevance of the feature values associated to that scenario. To make up for this limited number of examples Transductive SVM (TSVM) and Active Learning SVM (AL-SVM) are applied to *contrast* tagging as described in sections 5.4.3 and 5.4.4 respectively.

A factor affecting both precision and recall that we have already mentioned is that the training data set contains word pairs that are clearly contrastive when looking at text only but were not prosodically prominent in the conversation and so were not labelled as contrastive. Here is an example:

(e) I think that the judges should be left to do most of the sentencing, simply because, there is always a jury that might be swayed, by the moment, either to be too lenient or too vengeful, I guess.

where "lenient" and "vengeful" are contrastive but are labelled as a negative example in the data.

The presence of such examples in the training data makes the training data ambiguous (in the textual dimension) as it contains contradicting examples and so it affects

recall. It also affects precision because when the tagger classifies such examples as positive that classification is considered wrong. Switching the labels of all those examples would lead to better results and make the *contrast* labelling a proper detection task rather than a prediction task[6].

Perhaps the main problem for the identification of contrastive word pairs is the "wrong" scope of *contrast*. The training data contains several examples where two words were annotated as contrastive while actually the items related by the *contrast* relation are more than the two words as in the following examples:

(f) It seemed like the **further** I got away from Dallas, center, the more lot came with the house and the **lower** the price, at the same time.

(g) After **wasting** the first six years, partying and everything else, I decided, uh-huh, time to settle down and **do** something .

(h) Gosh, we**'re** keeping these men in prison for fifteen or twenty years on death row, and not **doing** anything with them.

(i) Until we **have** to learn to think that way, we **won**'t.

The annotated contrastive words are the two words mostly involved in the *contrast* but not the only ones.

This problem often occurs when the two annotated contrastive words are verbs, especially auxiliary verbs. Removing verbs from the training data reduces this problem and increases the accuracy of the tagger as shown in table 5.7.

Another negative factor affecting the tagger accuracy could reside in the correctness of the extracted features. For instance the extracted semantic features work on word senses but no word sense disambiguation is applied before extracting the semantic features so that the semantic features may refer to the wrong senses and consequently be wrong. Actually in experiment 1 (section 5.2) we used a word sense disambiguator but it turned out to be very time consuming and completely useless for the *contrast* labelling task.

[6]This problem may be seen as one of the aspects of prosodic variability, and thus as a problem that is intrinsic to the task of predicting prosodic marking of contrastive words and cannot be separated from it. However what we are claiming here and in more detail in the next sections and in the concluding chapter is that, in order to improve the tagger accuracy, we need to separate the pragmatic and "textual" component of the task from its phonetic part by first identifying the "textual" *contrast*, i.e., the pragmatic relation of *contrast* identifiable from text only, and then trying to predict whether the contrastive words (on text) will be prosodically marked or not.

Examples	d	j	Precision	Recall
All	2	2	74.08	24.4%
No verbs	2	2	77.27%	28.1%

Table 5.7: *Verbs and scope of* contrast. *Leave-one(sentence)-out evaluation of the fully automatic contrast tagger on all examples (All) and all examples except those in which the word pairs are verb pairs (No verbs).*

Other features like the features related to the syntactic dependencies may be not always correct because of the dependency parser can fail, especially on a type of text like the transcription of conversational speech which is usually much more "ungrammatical" than written text and so more prone to contain unusual syntactic structures never seen in the training data of the dependency parser.

Moreover, even if syntactic dependencies were always correct, the set of syntactic dependencies used, although it is the "standard" set of dependencies, would not be the most suitable for the *contrast* tagging task. For example a dependency that explicitly indicates that two words are related by a conjunction like the words "happy" and "calm" in "Are you happy or just calm?" would be more informative than knowing that "happy" and "or" are noun modifiers of "calm" as it happens when that sentence is parsed with MaltParser[7]. The dependency set proposed by Johansson and Nugues (2007), which contains dependency more directly linked to the concept of *contrast* and more specifically to the concept of syntactic parallelism, could be a more suitable set for *contrast* tagging.

The feature set of the *contrast* tagger seems to contain all the necessary information to detect examples of *contrast* where actually the tagger fails to recognize them. An example is:

(l) We threatened to make the other two make us dinner one time just to even it out since **we** seemed to be unfairly doing all the **cooking they** were doing all the **enjoying**

where *contrast* is mainly activated by syntactic parallelism which is largely covered by the feature set.

[7]The use of a specific dependency between two (or more) words linked by a conjunction has always been problematic for dependency parsing and for that reason it has been almost always avoided. As a consequence such syntactic link has been almost always represented by a non-specific and perhaps counter-intuitive dependency.

Again one possible explanation is the limited number of examples of some scenarios of *contrast*. An alternative or complementary explanation is that the feature set contains all the necessary information but, for some examples, the combinations of some features rather than the independent contribution of each single feature are determinant to identify *contrast* but the classifier fails to "recognize" these relevant combinations. This issue is discussed in more details in sections 5.4.1 and 5.4.2 where attempts to identify useful combinations of features are described.

Finally the approach to *contrast* labelling described so far assumes that the $\pm$*contrast* relation between two words does not depend on the $\pm$*contrast* relation of the two words with other words. This independence assumption can cause errors as in the sentence:

(m) I did a Sunday school lesson one time on the difference between the Old Testament and the New Testament where there 's a vengeful Lord in the Old Testament and there 's a loving Lord in the New Testament.

where in the leave-one(sentence)-out evaluation the classifier wrongly classifies as contrastive the pair consisting of the first "Old and the last "New" while the contrastive pairs containing the two words consist of the first "Old" and the first "New", and the last "Old" and the last "New".

The independence assumption can be relaxed by using a Conditional Random Field (see section 3.3.4) or a Markov Logic Network (Richardson and Domingos (2006)) based *contrast* classifier instead of the SVM based classifier proposed here. Relaxing the independence assumption might lead to better results but that is not guarantee as Conditional Random Field and Markov Logic Netweork may not be as good as SVM when dealing with large numbers of features as in our case.

5.4 Improving tagger accuracy

As we have seen in the previous section there are several problems affecting the accuracy of the *contrast* tagger. Some of these problems (e.g., lack of pragmatic "knowledge of the world") do not seem to have an immediate solution, given the NLP tools available at the moment, while other problems should be addressed by modifying the feature set (e.g., by modifying the set of dependency features) or modifying the approach to the task (e.g., to account for the problem of the scope of *contrast*).

Our attempts to improve the tagger accuracy are conservative in the sense that no further features are extracted but only the already available features and data (both

labelled and unlabelled data) are used.

The methods described in sections 5.4.1 and 5.4.2 aim to increase the *capacity* of the tagger, possibly without decreasing its generalization, by searching for predictive "combinations" of features that may be ignored in the "standard" SVM-based tagger. In several NLP tasks, new features are extracted by simply conjoining symbolic features. So, for example, the two features $POS(w_i)$ (POS of the current word) and $POS(w_{i-1})$ (POS of the preceding word) are combined in the new additional feature $POS(w_{i-1}) - POS(w_i)$. Usually the feature combinations are manually chosen. However, since we have used hundreds of features for *contrast* labelling, trying to manually select the most predictive feature combinations may be unfeasible or very time-consuming. To avoid such problem we use methods to automatically select predictive feature combinations.

The methods described in sections 5.4.3 and 5.4.4 aim to make up for the limited amount of training data. In section 5.4.3 Transductive SVM (Vapnik (1995)) is applied to make use of unlabeled data (i.e., unlabeled conversations), while in section 5.4.4 the labelled training data is slightly increased by applying Active Learning SVM (AL-SVM). AL-SVM is used to select and manually label a restricted number of particularly "useful" examples from the unlabeled data set.

5.4.1 Feature selection and combination

Adding (either manually or automatically) new features that are functions of features already present in the feature set can increase the capacity of a SVM-based classifier. In order to see that, suppose we have a training data set consisting of two binary features x_1 and x_2 and four points in the training data set as shown on the left side of figure 5.4, where the four points are not linearly separable in the 2-dimensional space (see the left side of figure 5.4).

However if a new feature x_3 where $x_3 = x_1 x_2$ is added to the feature set then the training data set becomes linearly separable as an additional degree of freedom which "frees" new hyperplanes (in a higher dimensional space) is added (see the right side of figure 5.4). Making the training data set linearly separable means increasing the capacity of the classifier because the accuracy of the classifier on the training data set increases (in the example of figure 5.4 from 50% to 100%).

When using polynomial kernels, increasing the order of the polynomial kernel corresponds to creating new features that are products of features from the original feature set. For example, switching from a linear kernel $k(\mathbf{x}, \mathbf{z}) = \mathbf{xz}$ to the following second-

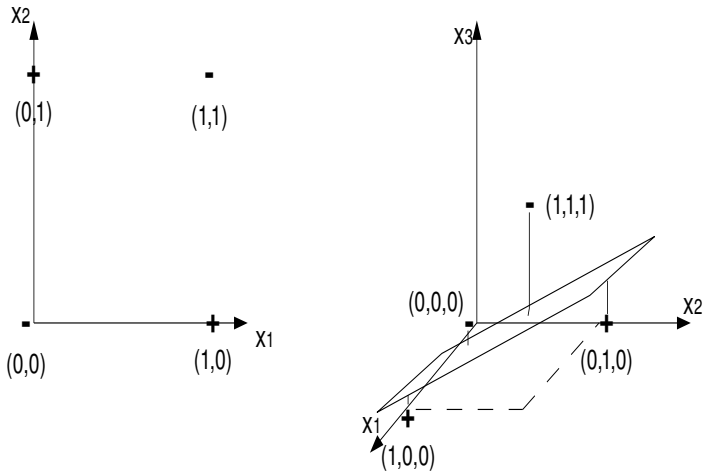

Figure 5.4: *Feature combination and SVM capability. On the left side of the figure the four data points are not linearly separable in the two-dimensional space defined by features* x_1 *and* x_2*. If a new feature* $x_3 = x_1 x_2$ *is added in the feature set the four points become separable in the new three-dimensional space (right side)*

order polynomial kernel:

$$
\begin{aligned}
k(1+\mathbf{xz})^2 &= (1+x_1z_1+x_2z_2)^2 && (5.8)\\
&= 1+2x_1z_1+2x_2z_2+x_1^2z_1^2+2x_1z_1x_2z_2+x_2^2z_2^2 && (5.9)\\
&= (1,\sqrt{2}x_1,\sqrt{2}x_2,x_1^2,\sqrt{2}x_1x_2,x_2^2)(1,\sqrt{2}z_1,\sqrt{2}z_2,z_1^2,\sqrt{2}z_1z_2,z_2^2) && (5.10)\\
&= \phi(\mathbf{x})^T\phi(\mathbf{z}) && (5.11)
\end{aligned}
$$

where the fifth feature in the new 6-dimensional space, $\sqrt{2}x_1x_2$, is a "combination" of the two original features. The "combination" is a product if at least one of the features involved is continuous-valued, while it is a conjunction if both features are binary.

Results in tables 5.5 show that the 2nd-order polynomial kernel is the best trade-off between generalization and capacity, while the first order polynomial has too little capacity and the 3rd-order has too much capacity.

The 3rd-order polynomial kernel creates more and longer (with 3 instead of 2 features combined) combinations of features that increase the accuracy on the training data but worsen the accuracy on the leave-one-out evaluation. Some of the combinations it creates may be good, i.e., combinations that increase capability without causing overfitting, while other combinations cause overfitting.

The method proposed here is based on the hypothesis that bad combinations are combinations that include "useless" original features, i.e., non-predictive features. If useless original features are removed than it could be possible to have "long" feature combinations (i.e., high order polynomial kernels) that increase the classifier capability without worsening its generalization.

In order to verify such hypothesis, first the feature set is reduced by means of a *feature selection* algorithm and subsequently polynomial kernels with order greater than 2 are used. The *feature selection* algorithm is a *filter*, that is a ranking algorithm that sorts by descending order all the features according to their value of symmetric uncertainty with the $\pm$*contrast* class (see section 3.5) and then reduces the feature set to the first N features, where N is manually set.

Table 5.8 shows the tagger accuracy for different values of N. The original feature set consists of 1645 features. For values of N ranging from 750 to 1250, results confirm the hypothesis that the use of polynomials with order greater than 2 on a reduced feature set outperforms the classifier trained on the original feature set. However the improvement in accuracy is disappointingly small.

Note that the selection of the feature subset described here does not take into account the machine learning method used. Alternatively or complementary to a *filter*, a

No. of Features	d	Accuracy	Precision	Recall
1645 (all)	2	97.23	75.00	20.73
50	2	96.80%	50.46%	22.36%
50	3	96.94%	55.89%	**23.17%**
250	2	97.06%	61.8%	22.36%
250	3	97.0%	59.53%	20.36%
500	2	97.13%	70.04%	20.32%
500	3	97.21%	73.24%	21.14%
750	2	97.18%	71.43%	20.32%
750	3	97.28%	76.39%	22.36%
1000	2	97.19%	71.84%	20.73%
1000	3	**97.29%**	76.71%	22.76%
1250	2	97.23%	75.0%	20.73%
1250	3	97.27%	**76.8%**	21.54%

Table 5.8: *Feature selection for* contrast *labelling. 10-fold cross validation. Results of classifiers using a forth order polynomial kernel are not shown as their accuracy is always worse than that of classifiers based on 2nd and 3rd order polynomial kernels.*

wrapper could be used. *Wrappers* are feature selection methods that consider the utility of a feature subset (and not a single feature) in terms of its impact on the classifier accuracy (on a validation data set). We have not used *wrappers* as they are computationally far more expensive than *filters*, but their use could further improve the *contrast* tagger accuracy.

5.4.2 Feature combination and selection

The *filter* method mentioned in the previous section is based on a correlation measure that considers the relevance, i.e., correlation to the class to predict, of a feature independently of all other features. As a consequence many selected features may be redundant while some excluded features can be actually relevant when "combined" with others. The method proposed in this next section tries to find useful conjunctions of features that may contain features that are poorly correlated with the *contrast*

variable and so no a-priori feature selection is applied.

In order to find conjunctions of features, all non-binary features must be transformed into binary features. The original feature set only contains a couple of non-binary features (that are in this case real-valued and $\in [0,1]$) so the binarization process is very limited. Then the feature combination and selection works as follows:

1. At each positive example search for all conjunctions of 2, 3, 4 and 5 features where all the features have value 1 (i.e., all features in the conjunction have value 1). These are the candidate conjunctions.

2. Compute the utility of each candidate conjunction, where the utility value is the difference between the number of times the conjunction occurs in a positive example and the number of times it occurs in a negative example

3. Remove all conjunctions with an utility value less a predefined threshold (set to 2 in the experiments) and sort all the remaining conjunctions in descending order of utility.

4. Add to the set of selected conjunctions the conjunction with the highest utility value

5. Remove all positive examples where the selected conjunction occurs and recompute the utility score of the candidate conjunctions on the new training data

6. If the set of candidate features is empty then stop, otherwise go back to step 3.

The set of candidate conjunctions extracted at step 1 might have included conjunctions of features that have value 0 (e.g., $conj_i = 1$ if $feat_1 = 1$ AND $feat_{34} = 0$ AND $feat_{106} = 1$) but that would lead to an intractably huge set of conjunctions[8].

When the set of conjunctions extracted with this feature combination and selection method are added to the original features set the classifier based on a linear kernel outperforms a classifier using the same kernel but the original feature set. However adding the set of conjunctions increases the capability of the tagger (accuracy on the training data is always higher than when the original feature set is used) but unfortunately it

[8] An alternative would be that of extracting conjunctions in the same way RIPPER extracts rules (and with rule validation and pruning deactivated). However, because of the way RIPPER extracts the rules, in that way the most correlated features would have more chances of being part of a conjunction than not relevant features, while the aim of the method proposed in this section is that of removing any bias that favors the most relevant (i.e., correlated) features.

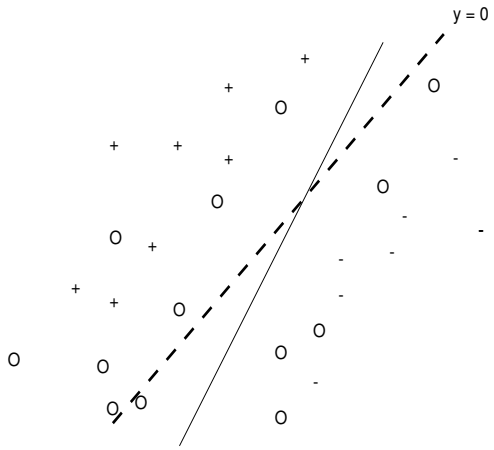

Figure 5.5: *Transductive SVM. The testing data points are added in the training data set as unlabeled data points (0).*

never outperforms the best original tagger (i.e., tagger with 2nd-order polynomial kernel and using the original feature set). Moreover, changing the threshold value and using different metrics to compute the utility score has not led to any improvement.

5.4.3 Transductive SVM for *contrast* labelling

The basic idea of Transductive Support Vector Machine (TSVM) (Vapnik (1998)) is that including unlabeled test data in the training data set makes the training data set a more accurate approximation of the real distribution of data points (i.e., that corresponding to an infinite training data set) around the testing data. If the testing data points are close to the decision boundary a consequently more accurate approximation of the distribution around the decision boundary can lead to a better classifier. In fact unlabeled data may be useful if they fall within the *margin*, where the *margin* is the *margin* found when only labelled examples are taken into account. As it is shown in figure 5.5, knowing where unlabeled data points are may lead to a better decision boundary. The TSVM implementation proposed by Joachims (1999b) was evaluated for *contrast* classification again in a leave-one(sentence)-out setting. As it is shown in table 5.9, TSVM produced poor results, with a substantial improving in recall but a drastic drop in precision which we find hard to interpret. A possible explanation is

TSVM Type	Accuracy	Precision	Recall
No TSVM	97.30%	76.62%	23.89%
TSVM	95.61%	31.53%	30.77%
Pseudo-TSVM (+1 conversation)	97.32%	77.63%	23.89%
Pseudo-TSVM (+2 conversation)	97.22%	66.67%	28.34%
Pseudo-TSVM (+3 conversation)	97.10%	64.37%	22.67%

Table 5.9: *Evaluation of the TSVM* contrast *tagger. Leave-one(sentence)-out evaluation. The No TSVM is the original SVM based predictor which does not use unlabelled examples. The TSVM is the TSVM based tagger in which the unlabeled data set only consists of the unlabeled testing data points. In the Pseudo-TSVM taggers the the unlabeled data points are not from the testing data set.*

that in the TSVM algorithm used (see Joachims (1999b)) the unlabeled data points are assigned a class within the constraint of having in the unlabeled data set the same ratio of positive and negative examples of the (original labelled) training data. Since the TSVM tagger has better recall but lower precision than the normal SVM tagger it might be possible that this constraint forces some of the testing data set consisting of examples from the longest sentences (in the leave-one(sentence)-out evaluation) to have too many positive examples.

Note that *transductive learning* contrary to *inductive learning* "adapts" the placement of the decision boundary to accommodate the distribution of the testing data and so it searches for a local solution, which changes when the testing data change, instead of a global solution, which is independent of the testing data used.

It is possible to slightly change the TSVM approach by including in the training data set unlabeled data that do not belong to the testing data set. Doing that we still get some useful information from the unlabeled data about the real data distribution but instead of searching for the best local solution we try to improve the global solution.

Table 5.9[9] shows the tagger accuracy with TSVM when unlabeled examples from one, two and three unlabeled conversations are added to the training data. The three

[9]Note that in this table the reported accuracy of the standard SVM tagger is slightly different from that reported in 5.5 for the same tagger. The little difference is due the fact that after a small bug fixing, it turned out that the training data set contains 247 positive examples instead of 246. This new training data set was used to evaluate the TSVM and the Active Learning SVM tagger but not the taggers of the previous sections

conversations contain a number of examples which is comparable with that of the original training data set but have a much lower ratio between positive and negative examples.

5.4.4 Active Learning SVM for *contrast* labelling

Active Learning may allow to save a lot of annotation time. Instead of manually labelling a large set of unlabeled examples and training a classifier on it, through Active Learning it is possible to build an almost equally good classifier trained on a much smaller subset of training data points. Active Learning SVM for binary classification works as follows:

1. The classifier is trained on an initial training data set consisting of few labelled examples (usually just one positive and one negative)

2. A query function uses the classifier to find in the unlabeled data the "best" example(s)

3. The selected example(s) is manually labelled and added to the training data set (the example(s) selected are presented to the annotator in the form of the sentence containing the two words and with the two words highlighted)

4. The classifier is re-trained on the new training data set

5. If the stopping criterion is met the AL-SVM process stops, otherwise it continues from step 2.

In our implementation of AL-SVM for *contrast* labelling the initial labelled training data is the original training data set used in section 5.3 while the query function consists in selecting the examples that are closest to the decision boundary. As a consequence such a query function selects the most informative examples, that is the examples for which the classifier is less confident.

That can be shown graphically (figure 5.6). Once labelled and added to the old previous training data set, examples close to the decision boundary are more likely to displace the decision boundary after re-training than examples far from the decision boundary. Other query functions have been proposed in the literature (most notably by Tong and Koller (2002)) but they are generally much more time consuming and in most of the cases do not lead to significantly better results.

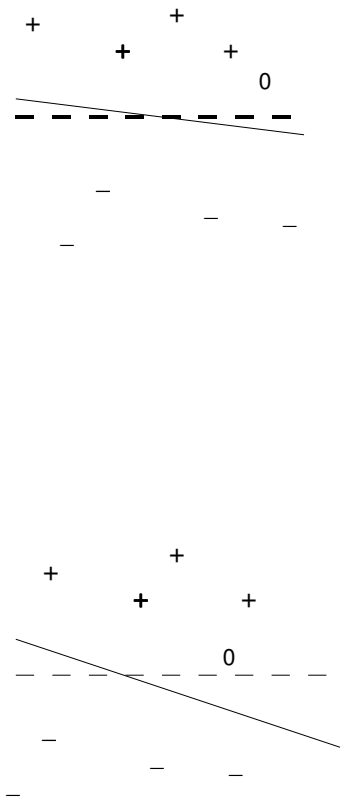

Figure 5.6: *Active Learning SVM. The closer the queried data point (0) is to the initial decision boundary (dotted line) the biggest the displacement of the decision boundary. The continuous line is the decision boundary after retraining (if the queries data point is labelled as positive). The figure shows two distinct examples, in the bottom example the displacement of the hyperplane is larger than on the top example.*

The number of examples selected at each training round was fixed to 25. Selecting more than one example at each loop raises the issue of the redundancy of the selected examples. Two or more examples can be both very close to the decision boundary and also be very similar (i.e., close) to each other and so redundant for classification purposes. To avoid such redundancy the metric function should include a measure of the independence of the feature vectors associated to the example, however, as in most of previous work on AL-SVM, such measure has not been included in the query function.

An important issue in AL-SVM concerns the stopping criterion. Vlachos (2008) proposes to compute after each re-training on some held-out data the average distance of the examples from the decision boundary. The average distance is a measure of the confidence of the classifier. The smaller the distance the less confident the classifier is. At the beginning of the Active Learning process the average distance (i.e., the confidence) increases as informative data points are added to the training data set. At some point the confidence will reach an asymptotic value or a maximum peak followed by a decrease depending on whether the last data points added to the training set have non influence on the classifier or are noise that confuses the classifier.

We have used the confidence measure to track the evolution of confidence of the *contrast* classifier at each learning loop, but it has not been used for the stopping criterion. Instead the trend of the classifier accuracy on the testing data is monitored after each training loop and AL-SVM stops when accuracy constantly decreases.

Finally note that the queried data points were labelled without listening to the correspondent speech. As a consequence the additional training data is different from the original in that it contains example of contrastive word pairs (in the "textual" dimension) not necessarily prosodically marked. All queried data points were labelled by us.

To evaluate the potential advantage of adding new training data through AL-SVM a leave-one(sentence)-out evaluation is carried out as in section 5.4.3 for the Pseudo-TSVM. The new data points queried by the AL-SVM module are only used in the training data and never used as testing data, so the number of left out sentences is the same as in the leave-one(sentence)-out evaluation in experiment 2 (section 5.3). Results are shown in table 5.10.

Unfortunately the use of AL-SVM does not produce any improvement and even produces a small decrease in accuracy although the tagger confidence increases. The small decrease in accuracy is due to a decrease in recall, while the precision increases

No. of loops	Accuracy	Precision	Recall	Confidence
0	97.30%	76.62%	23.89%	1.273
1	97.28%	78.26%	21.86%	1.275
2	97.27%	78.79%	21.05%	1.274
3	97.26%	78.46%	20.65%	1.273
4	97.24%	78.13%	20.25%	1.286
5	97.22%	75.76%	20.25%	1.287
6	97.20%	73.91%	20.65%	1.292
7	97.26%	80.32%	19.84%	1.294
8	97.22%	79.31%	18.62%	1.295

Table 5.10: *Evaluation of the AL-SVM based* contrast *tagger. Leave-one(sentence)-out evaluation in which only the data points from the sentences of the original training data set (loop 0) are tested. At each loop, 25 new data points are queried from the AL-algorithm, then labelled and added to the training data set. The confidence is the average distance from the decision boundary of a set of data points (which belong neither to the training data set nor the testing data set).*

(with the exception of loop 5 and 6) with respect to the standard SVM tagger. We believe that the main reason of a decrease in recall and an increase in precision is a consequence of our choice of being very conservative when labelling the queried data examples. In a lot of the queried examples classified as positive the two words were only part of two contrastive phrases. We labelled all those cases as negative apart from few exceptions (where actually the scope of *contrast* was not very clear to us).

One of the possible reasons why AL-SVM fails to increase recall is that, an AL-SVM algorithm like the one we used searches for data points that are problematic for the SVM classifier but ignore the real distribution of the data points. As Dasgupta and Hsu (2008) point out, if an Active Learning method ignores the data distribution it may miss data points that are very important for classification. For instance in the *contrast* classification task, if the unlabeled example "your-mine" from sentence:

(a) Your attitude is opposite from mine

had no feature values indicating the possible presence of *contrast* because the activation of *contrast* is almost entirely due to the bigram "opposite from" which never

occurs in the training data, then the example would be classified with high confidence as negative since most of the training data points are negative.

We expect that such a problem would be reduced if the new training data set for *contrast* tagging, created using AL-SVM, were built from an initial training data set consisting of one positive and one negative example. A balanced initial data set would not bias the tagging to a strong "preference" for negative examples and so for examples like "your-mine" in sentence (a) the classifier would have a very low classification confidence.

Although AL-SVM has brought no improvements in *contrast* tagging still some advantages come from Active Learning. One advantage is that also examples not occurring within sentences containing *contrast* are included in the training data set. In section 5.3 the training and testing examples only came from sentences containing at least one contrastive pair, all sentences not including contrastive pairs are not "represented" by any example. As a consequence the ratio of positive and negative examples does not reflect the real ratio which should count many more negative examples. A valid justification for only including negative examples from sentences including contrastive pairs is that those negative examples are generally closer (in the feature space) to the positive examples than all other negative examples, in that they generally share many more feature values with the positive examples and so are more useful for finding the best decision boundary. Including all negative examples from sentences that do not contain any contrast pair (either word or phrase pairs) may produce a very noisy data with a lot of word pairs that are actually contrastive when looking at text only but were not prosodically marked. However the use of AL-SVM should help including useful negative examples and produce a more realistic ratio.

Finally a further very important benefit from AL-SVM is that AL-SVM can be used to quickly build annotated training data sets from other types of corpus.

5.5 Other Data?

Being a novel task the automatic labelling of *contrast* presents several challenges. In section 5.3.1 we have seen all the main problems and their possible solutions that have to be addressed in order to improve the accuracy in *contrast* tagging.

Some of the problems are due to the fact that the corpus used is a corpus of conversational speech, which by nature is rich in ungrammatical constructions that make feature extraction problematic (especially the extraction of syntactic features) and, with

respect to corpora of written text, patterns of *contrast* harder to identify.

Obviously a corpus of conversational speech is not a problem per se but it would be very interesting training and testing the *contrast* tagger on different corpora with different registers. To the best of our knowledge the corpus we used is the only available corpus in which *contrast* is annotated, but it may be possible to quickly annotate *contrast* on a different corpus, like the Wall Street Journal on the Penn Treebank (Marcus et al. (1993)), by using the AL-SVM implementation for *contrast* classification.

In a corpus like the Penn Treebank we should remove the constraints on the contrastive words on being prosodically marked as the corpus is a collection of newspaper articles. As consequence the task would slightly change in the labelling of *contrast* that is identifiable from text and that may or may not be prosodically marked. In this new task the annotated training data would not contain the contradicting examples (on the "textual" dimension) of the training data we have used. Moreover we expect that in a more "grammatical" corpus those scenarios of *contrast* that are mainly activated by syntax are easier to identify as they repeat more regularly and without the "interference" of ungrammatical phrases. In such a context the use of new features like, for instance, *dependency tree kernels* (Culotta and Sorensen (2004)) may improve the tagger accuracy since they allow to compute the similarity of two examples where the similarity is mainly similarity of the syntactic structures (i.e., trees) of the two clauses containing the two examples[10].

5.6 Summary

In this chapter we have addressed the problem of the automatic labeling of *contrast* where *contrast* is meant as the relation linking two words that contrast with each other or two words in which one contrasts with the other. We have shown how the training data were collected and described and motivated the training feature set. We have shown the accuracy in *contrast* tagging when using a SVM-based *contrast* classifier. Results show that our novel approach to *contrast* tagging allows a good precision (around 80%) in *contrast* identification, although the recall (just below 25%) is not as much satisfactory.

Results also show that only an approach that combines semantic and syntactic fea-

[10] Actually one of the main reasons we decided to build a SVM-based tagger is the possibility of using *tree kernels*. *Tree kernels* would allow to "memorize" the syntactic structures in which *contrast* usually occurs. However in our view the use of such kernels would not make that much sense in the Switchboard corpus

tures allows to identify contrastive words with high precision , a high precision that is essential for TTS applications (and makes our classifier a suitable *contrast* tagger for TTS applications) and cannot be achieved by using the textual features proposed in previous work on the identification of contrastive words.

In the analysis of error of our *contrast* tagger we have identified several factors affecting the tagger accuracy. In order to tackle some of them we have proposed methods that range from feature selection and "combination" to Active Learning SVM. Unfortunately such "enhancing" methods have failed to significantly improve tagging accuracy in the Switchboard corpus. One of the main obstacles to a very accurate *contrast* tagging in the Switchboard corpus is the widespread "ungrammaticality" of the corpus. We expect our tagger and its "enhancing" methods to be more successful on more "grammatical" corpora. Since at the moment there is no *contrast* annotation in corpora other than Switchboard, the manual annotation of *contrast* in other corpora could greatly benefit from our implementation of Active Learning SVM for *contrast*.

Chapter 6

Generating Prosodic Prominence: Experiments in Text-to-Speech Synthesis

The previous chapters concerned the prediction of prosodic prominence events from text. The next step consists in assessing the utility of such predictions in TTS synthesis through large scale perceptual tests.

The experiment described in section 6.2 investigates the utility of accurate pitch accent prediction in HMM-based TTS synthesis while section 6.3 describes and evaluates a method to generate contrastive accents on automatically detected contrastive word pairs.

As discussed in section 4.5 not only the predictions of a predictor but also its confidence (uncertainty) in prediction could be useful information to be integrated in a TTS system. This is the issue addressed in next section.

6.1 Experiment 1: Including Pitch Accent Optionality in Unit-Selection Text-to-Speech Synthesis

Goal of the experiment described in this section[1] is testing the hypothesis claimed in section 4.5. That hypothesis says that including the "uncertainty" of a pitch accent predictor, which is strongly correlated with pitch accent optionality, in a target cost function including pitch accent in the linguistic feature set, can improve both segmental and prosodic quality of the synthetic speech.

Integrating the predictor's confidence in the target cost makes the constraints on the accenting value of the candidate speech units dependent on the actual necessity of having a syllable accented or not. Thus when the predictor is very uncertain of its output the constraints on the candidate units are relaxed so that the search space of the unit selection module is enlarged and higher are the chances of selecting a good sequence of units without worsening the generation of prosodic prominence.

In order to test our hypothesis we compared a TTS system having the modified target cost function of equation 6.1 with one having the standard target cost function (i.e., a function where the uncertainty of the accent predictor is always zero, $H_p(w_i) = 0$ for any w_i). We call the former system EWC (which stands for Entropy Weighted Cost) and the latter SC (standard cost).

The target cost function T_f for the pitch accent feature in the EWC system is defined as:

[1] Joint work with Rob Clark and Volker Strom. Rob helped to design the perceptual test (other than actively supervising my work as for the rest of this thesis work) and Volker to implement the new target cost function

$$T_f = \begin{cases} 0 & \textit{if } s_t[f] \textit{ and } u_t[f] \textit{ are equal} \\ 1 - H_p(w_i) & \textit{otherwise} \end{cases} \tag{6.1}$$

where $s_t[f]$ and $u_t[f]$ are the accent values ($\pm$accent) of the target unit and the speech unit respectively, w_i is the word of the two units, and H_p, the predictor uncertainty, is defined as follows:

$$\begin{aligned} H_p(w_i) = \quad & -\log(P(y_i|\mathbf{x_i}))P(y_i|\mathbf{x_i}) \\ & -\log(1-P(y_i|\mathbf{x_i}))(1-P(y_i|\mathbf{x_i})) \end{aligned} \tag{6.2}$$

where $P(y_i|\mathbf{x_i})$ is the probability of the predicted accent value y_i and $\mathbf{x_i}$ is the vector of training features.

The target cost function for the pitch accent feature (T_f) of system SC can only have value 0 (if $s_t[f]$ and $u_t[f]$ are equal) or 1 (if $s_t[f]$ and $u_t[f]$ are different).

We expected system EWC to outperform system SC in a large scale listening test. Again note that our expectations were based on the assumption of the independence of pitch accent placement discussed in chapter 3 and 4.

6.1.1 Implementation details

The unit selection TTS system we used is the version of Multisyn Festival (Clark et al. (2007)) described in Strom et al. (2007), where "standard" and "emphatic" pitch accents are linguistic features of the target cost function. We only made two modifications to that version, one concerning the target cost function and one concerning the pitch accent predictor used for the prediction of pitch accent placements in the test sentences. Note that we did not modify the pitch annotation of the speech database. The pitch accent predictor used to annotate the speech database is described in Strom et al. (2007). It does not use acoustic features.

In the EWC system the T_f function for the pitch accent feature was modified according to equation 6.1. In both EWC and SC the pitch accent predictor was substituted with the CART(Wagon) predictor described in 3.3.1 using POS, Information Content and Relative Information Content as training features. We did not use the whole training feature set as this experiment was carried out before creating the final pitch accent predictor of chapter 3.

A unit (which is a diphone in Multisyn Festival) is considered accented if it belongs to a vowel of a stressed syllable in an accented word or if it precedes a vowel of a stressed syllable in an accented word (like in Strom et al. (2007)). The choice of this

context assumes that accenting has a phonetic effect on the nucleus of the stressed syllable of the accented word and on the phone preceding it.

The target cost associated to the pitch accent feature is considerably higher than those associated to the remaining features in order to give high importance to prosodic prominence during the unit selection process. This choice allowed to obtain an improvement in speech quality with respect to a TTS system not including the pitch accent feature (Strom et al. (2007)).

Among all the other target cost features the only feature somehow correlated to pitch accenting is the POS feature.

The speech database used is the same used in Strom et al. (2007).

6.1.2 Test design

To create a set of utterances for evaluation we ran our pitch accent predictor on a subsection of the BURN corpus and of the Herald news (which is the same corpus used for language modelling in chapter 3) on sentences no longer than 20 words. The predicted accent value and its associated H_p value were assigned to each word and the sentences were ranked according to the average value of H_p per each word from the highest to the lowest. Since the "standard" system assumes H_p to be always zero (i.e., prediction uncertainty is not taken into account) and since we wanted to select the sentences where the target costs of the two systems were most different we selected for the listening test the first 15 sentences having the highest rank (and producing audible differences between the two systems).

Each sentence was synthesized using both TTS systems so 15 pairs of utterances were generated. Each pair was presented to each participant twice but in reversed versions (i.e., EWC-SC and SC-EWC) so each participant listened to a total of 30 pairs whose order was randomized per each participant.

The listening tests were carried out through a web browser and, for each utterance pair, the participants could either express a preference for one of the two utterances or no preference.

46 subjects were recruited, all of them are native English speakers. All participants were paid. All participants used headphones. The tests lasted approximately 15 minutes each.

	EWC	SC	No-preference	p-value_1	p-value_2
All pairs	587	439	354	$p < 0.00001$	$p = 0.00007$
Consistent preferences only	382	206	156	$p < 0.00001$	$p < 0.00001$

Table 6.1: *EWC vs. SC. In the* All pairs *row the comparison is made on all the pairs* $(30 * 46)$ *presented in the experiments. In the* Consistent preferences only *row, the comparison is made only on the pairs where the preferences of the subjects where consistent, i.e., when subjects expressed the same preference in the two pairs having the same utterances but in reversed order. Columns 2, 3 and 4 report the number of preferences for the three options. The p-values are from two-sided Binomial tests. The p-value_1 is computed by excluding the* No preference *choices from the overall set of choices, while the p-value_2 is computed by splitting the* No preference *set into two halves and summing one half to the EWC preferences and the other half to the SC preferences.*

Sentence ID	EWC	SC	No-preference
Sentence 2	23	53	16
Sentence 3	78	8	6

Table 6.2: *Best cases for EWC and SC. Sentences with highest number of preferences for EWC and for SC respectively.*

6.1.3 Results and discussion

The overall results are shown in Table 6.1. We computed the number of preferences for the three options (EWC, SC and No preference) on the overall set of pairs, and on the set only containing pairs where the subject's choices were consistent, that is were the subject chose the same option in the two pairs having the same utterances (but in reversed order). We then ran two different kinds of two-sided Binomial tests: one where all the preferences for the "No preference" option were excluded and one where they were split into two equal halves and one half was summed to the EWC preferences and the other one to the SC preferences.

All conditions and tests show a statistically significant preference for EWC, with p-values far below 0.001.

Looking at each single sentence the difference between the two systems is less evident since for only 5 out of 15 sentences there is a significant (p-value_2 < 0.01) preference for EWC, whereas for 4 sentences the significant preference is for system SC, and for the remaining 6 sentences there is no significant preference for either of the two systems.

Despite this small difference between EWC and SC at the utterance level the overall results show a significant preference for EWC because, when significant, the preferences for EWC are more definite than the preferences for SC (see Table 6.2).

This behavior is also observable when looking at the "consistent" results (last row of table 6.1). The proportion of consistent evaluations is quite low (54%) indicating that the subjects in general found it hard to hear clear differences between the two utterances (or that the subjects were not capable of expressing a preference), however when the subject's choices are consistent (which is a consequence of a clearer difference between the two utterances) the preference for system EWC is more evident.

This fact might imply that it is in the critical cases, where unnecessary strict constraints on prosodic prominence heavily damage the synthetic speech quality, that including the accent predictor uncertainty helps and produces a clearly perceivable improvement. When there are not critical cases the contribution of the prediction uncertainty becomes almost irrelevant.

Listening to the test utterances we noticed that in a couple of cases what we perceived was the opposite of what we would have expected: where the value of H_p for a given word was high, that word was strongly accented by the EWC system and not accented or slightly accented by the SC system. We believe this behavior is mainly due to: 1) the intrinsic "instability" of the unit selection technique; 2) the inaccuracy of the pitch accent annotation in the speech database.

We did not compute the annotation accuracy but we know that at most a pitch accent detector has an accuracy around 85%, an accuracy that may be not good enough for this task and that may have significantly affected our results. We know that because of the intrinsic variability in accent annotation (see discussions at the end of sections 2.3 and 4.1), which causes a certain degree of disagreement between annotators, aiming for a 100% accuracy in accent detection does not make sense, so we do not know exactely the gap that still exists between automatic and human accent detection[2].

However, even if a 85% accuracy were a human-like accuracy, the accuracy of the

[2]Actually we may want automatic annotation to be better that human annotation, i.e., to be more consistent

accent automatic detection on the speech database used in this experiment is certainly below that value since the voice of the speech database and that on which the pitch accent predictor was trained are different (the predictor was not trained on one single voice but on SWBDC and BURNC voices). Perhaps only a manual annotation of the speech database would significantly reduce the number of unexpected outputs.

6.2 Experiment 2: Pitch accents in HMM based Text-to-Speech synthesis

Goal of the experiment described in this section[3] is investigating whether accurate information about pitch accent placement gives any benefit to HMM-based TTS synthesis.

Instead of comparing two HMM based systems, one having pitch accent as one of the linguistic features and the other one having the same feature set but no pitch accent feature, we compared a system (henceforth system HTS05) using a poor performance predictor with a system (henceforth HTS05-PP) using a much better predictor and additional features related to prosodic prominence.

Such comparison aims to find out whether improvements in pitch accent prediction leads to a better prosodic prominence modelling in HMM speech synthesis or this effort to improve pitch accent prediction brings no actual advantages.

Note that the accent prediction does not only concern the prediction of accent placements in the test sentences but also the prediction (actually detection) of the accents in the training speech data.

6.2.1 Implementation

System HTS05 is the HMM based TTS system described in Zen and Toda (2005). It uses the pitch accent predictor integrated in Festival (which is not the one used in Strom et al. (2007)) whose accuracy on the BURN corpus is very poor (below 70%) as reported in Sridhar and Bangalore (2008).

System HTS05-PP uses the state-of-the-art pitch accent predictor described in chapter 3, i.e., bagging-based predictor with full feature set. The predictor tested on the BURN corpus has a much higher accuracy (above 85%).

[3] Joint work with Sebastian Andersson and Junichi Yamagishi. Sebastian and Junichi helped to build the TTS systems

Both accent predictors were used to labelled the training data and the test sentences.

The two systems not only differ in the accent predictor used but also in the linguistic features that are correlated to prosodic prominence. In HTS05 those features are:

- (stress-dependent) name of current syllable
- {previous,current,next} syllable stressed/not-stressed
- {previous,current,next} syllable accented/not-accented
- distance (in number of syllables) to {previous,next} stressed syllable
- distance (in number of syllables) to previous,next accented syllable
- POS of previous,current,next word ("content word" is one of the values)
- distance (in number words) from {previous, next} content word
- length (in syllables) of {previous,current,next} word

All other features are mainly positional features (e.g., "position of syllable in the word containing it", "position of word in the prosodic phrase containing it") and "length" features (e.g., "length (in number of words) of the sentence"). As some of them convey information on the (possible) prosodic structure of the utterance they might also convey some information on the prosodic prominence pattern of the utterance.

In HTS05-PP the features related to prosodic prominence are:

- {previous,current,next} phoneme accented/not-accented
- {previous,current,next} syllable accented/not-accented
- current word accented/not-accented
- uncertainty of the pitch accent predictor (only at the word level)
- most of the training features extracted for the pitch accent predictor (see chapter 3):
  - POS of {previous,current,next} word (extracted with a more accurate POS tagger than that used in system HTS05)
  - Information Content of {previous,current,next} word
  - Relative Information Content of {previous,current,next} word

- Inverse Relative Information Content of {previous,current,next} word
- Length (in number of characters) of {previous,current,next} word
- Information Content of Concepts of {previous,current,next} word
- Syntactic Dependencies of {previous,current,next} word
- Cached Information Content of {previous,current,next} word

The Dependencies-based Relative Informativeness (DRI) feature was not included as it turned out to be not useful for pitch accent prediction.

The training features of the pitch accent predictor were included to take into account intra-speaker variability in pitch accent placement. In fact the pitch accent predictor was trained on an American English voice, the f2b voice in the BURN corpus, while the voice on which the HMM-based system was trained is a British English voice, the voice Roger from the Blizzard 2008 speech data set (Karaiskos et al. (2008)). The best accent predictor for f2b may not be the best accent predictor for Roger. Including the training features of the predictor in the linguistic feature set of the TTS system partially makes up for this discrepancy as they are the features most correlated to pitch accenting but are speaker-independent.

Some of the training features of the pitch accent predictor were included also with the aim of "capturing" weak-strong prominence relations between two adjacent words that cannot be captured using the accent feature only.

The "uncertainty of the pitch accent predictor" is included because it is correlated with accent optionality which can be in turn correlated with the acoustic realization of a pitch accent (i.e., "highly optional" accents might have a different distribution of the acoustic coefficients with respect to "compulsory" accents)

The "phoneme accented/not-accented" is a feature that is not included in the feature set of system HTS05. A phoneme is accented if it is the nucleus of an accented syllable.

Since accenting has no or very weak phonetic effect on non-accented words surrounding accented words (see Turk (1999)), in system HTS05-PP the next (or previous) syllable (and phoneme) is accented only if it belongs to the same word of the current syllable (phoneme). There is no such constraint in system HTS05.

According to this choice, in system HTS05-PP there are no features indicating the distance to {previous,next} accented syllable and to {previous,next} content word.

Both systems have the syllable-level feature "name of syllable nucleus" (and "name of phoneme"). However in system HTS05-PP the same syllable nucleus has two different names depending on whether it is accented or not (so the feature actually becomes

"accent dependent name of syllable nucleus"). This distinction is intended to capture possible specific phonetic effects of accenting since accenting might have different phonetic correlates depending on the syllable nucleus that is accented. This choice is consistent with the "stress-dependent name of syllable" used in both systems.

The speech data consists of all utterance from the whole Arctic section and part of the newspaper section of the Blizzard 2008 speech data (Karaiskos et al. (2008)). The speech data was split in 2025 utterances for training and 30 for testing.

Both systems were trained using the Eddie grid engine (Richards and Baker (2008)).

6.2.2 Test design

Two types of listening tests were carried out. One is the same preference test of experiment 1 where participants were presented utterance pairs and could express a preference for one of the utterance or no preference. We synthesized all 30 test sentences using both systems but only 12 utterance pairs were used in the listening test. Since, from a first listening, the differences between the two systems turned out to be on average very small we selected the utterance pairs where the differences between the two utterances were most perceivable. We could not use all 30 utterance pairs because the participants had also to carry out the listening test described in section 6.3 and so we needed the listening tests to be not too long (and not too expensive).

The other test is a similarity test in which participants listened to each of the 15 utterance pairs of the preference test plus a corresponding natural voice utterance and had to indicate which of the two synthesized utterances sounded most similar to the natural utterance. If the two synthesized utterances sounded equally similar to the natural utterance the participants chose the "both" option.

While in the preference test each utterance pair was presented in both orders (e.g., A-B and B-A) in the similarity test it was presented only once (the order was randomly chosen).

The motivaton behind the similarity test is that it allows to see if a more detailed model of prosodic prominence (like that of system HTS05-PP with respect to system HTS05) allows to better capture specific traits of the training voice. A preference test does not necessarly allow to observe that.

30 British English native speakers were paid to participate to the (preference and similarity) listening test which lasted approximately 15 minutes. All participants used headphones.

6.2.3 Results and discussion

The overall results of the preference test are shown in table 6.3. There is no statistically significant preference for either of the two systems. Contrary to our expectations the HTS05 system even slightly outperforms system HTS05-PP. Surprisingly, using a much better pitch accent predictor and some speaker-independent features highly correlated to accenting does not improve the speech quality of HMM based speech synthesis.

This results contradict what we see when observing the clustering trees for duration and F0 of the two systems. In system HTS05-PP the feature ”current phoneme accented” is on top of the clustering trees meaning that this feature is very important to cluster speech segments in the acoustic space. Also other features related to pitch accenting are in high positions of the trees. On the contrary, in the clustering trees of system HTS05, features related (directly and indirectly) to pitch accenting are far from the top nodes.

We might assume that although system HTS05-PP fails to outperform HTS05 in the preference test, it still might be possible that its more accurate accent prediction and its linguistic feature set allows a more detailed linguistic/symbolic prosodic modelling of the original natural voice and so leads to the generation of patterns of prosodic prominence that are closer to those of the original natural voice but not perceived as better than those generated by system HTS05.

Nevertheless results from the similarity test (table 6.4) show that HTS05-PP does not generate utterance that sound more similar to the natural utterance than those generated by HTS05.

In conclusion this experiment has shown that improving pitch accent prediction does not give any benefit to a HMM based TTS system and that a “bad” pitch accent predictor and a gross distinction between function and content work as well as a state-of-the-art pitch accent predictor.

As we will discuss in more detail in the concluding chapter of this thesis, our hypothesis is that HTS05-PP failed to improve the prosodic prominence realization of the HMM-based TTS system because its prosodic prominence model, which mainly relies on the simple distinction between accents and non-accents, is an oversimplistic model of prosodic prominence that “over-flattens” the complex hierarchical structure of prosodic prominence.

Obviously such hypothesis needs further investigations before being validated. At

	HTS05-PP	HTS05	No-preference	p-value_1	p-value_2
All pairs	212	227	281	$p = 0.85$	$p = 0.63$
Consistent preferences only	84	98	158	$p = 0.41$	$p = 0.55$

Table 6.3: *HTS05-PP vs. HTS05 - preference test. In the* All pairs *row the comparison is made on all the pairs ($24 * 46$) presented in the experiments. In the* Consistent preferences only *row the comparison is made only on the pairs where the preferences of the subjects where consistent, i.e., when subjects expressed the same preference in the two pairs having the same utterances but in reversed order. Columns 2, 3 and 4 report the number of preferences for the three options. The p-values are from two-sided Binomial tests. The p-value_1 is computed by excluding the* No preference *choices from the overall set of choices, while the p-value_2 is computed by splitting the* No preference *set into two halves and summing one half to the HTS05-PP preferences and one half to the HTS05 preferences.*

the moment it is based on an experiment carried out on one voice only. Most importantly, because of prosodic variability and since the predictor was trained on a voice different from that on which the TTS system was trained, the state-of-the-art predictor may still be not good enough in detecting the actual pitch accent placements in the training speech data. We might include acoustic feature in the feature set of the pitch accent predictor (which in that case would become a pitch accent detector) but as we have seen in section 2.4 that probably would not produce any significant improvement. Perhaps more definitive conclusions on the utility of pitch accents could be drawn if pitch accents were manually annotated in the training speech data.

HTS05-PP	HTS05	Both	p-value_1	p-value_2
120	122	118	$p = 0.95$	$p = 0.96$

Table 6.4: *HTS05-PP vs. HTS05 - similarity test. The first two columns show the number of cases in which one of the two utterances was judged most similar to the natural-voice utterance. The third columns shows the number of cases were the two utterances were judged equally similar to the natural-voice utterance*

6.3 Experiment 3: Generating Emphatic *Contrast* in HMM based Text-to-Speech Synthesis

From the previous experiment it turned out that an accurate pitch accent prediction/detection is not essential for prosodic prominence generation in HMM-based speech synthesis. We now want to investigate whether going beyond the usual ±accent distinction leads to any benefit in the generation of HMM-based speech synthesis. Goal of the work[4] presented in this section is generating an appropriate prosodic marking of contrastive words in a HMM-based TTS system.

This work follows a previous attempt we made to generate contrastive accents (Badino et al. (2009)). In that attempt the contrastive accents turned out to be too emphatic so that a "standard" pitch accent on contrastive words (i.e., contrastive when looking at text only) was generally preferred to the contrastive accent.

An interesting result that came up from Badino et al. (2009) is that emphatic contrastive accents on non-contrastive (but having same POS) word pairs are much less acceptable than emphatic contrastive accents on contrastive words. Although this could look as a pretty obvious result it empirically justifies a *contrast* tagger that detects the *contrast* relation rather than a tagger of contrastive words that ignores which word contrasts which word.

To the best of our knowledge the work presented here and in Badino et al. (2009) are the first attempts of generation of contrastive accents in HMM-based speech synthesis.

6.3.1 Implementation

The *contrast* tagger described in chapter 5 can be used either to detect contrast during synthesis time (i.e., when the TTS system generates speech) and off-line to collect examples of *contrast* in speech. The examples could be added to the speech database of the TTS system or/and used to analyze the prosodic realization of *contrast* in several different "scenarios" of *contrast*.

In both unit-selection and HMM-based speech synthesis we need examples of *contrast* to store *contrast* (in unit-selection speech synthesis) or "learn" *contrast* (in HMM-based speech synthesis).

[4]Joint work with Sebastian Andersson, Rob Clark and Junichi Yamagishi. Sebastian and Junichi helped to build the TTS system while Rob helped to design the perceptual test.

Collection of *contrast* in speech may be a necessary step to generate contrastive accents in speech synthesis (although it does not guarantee a good modelling of contrast because of limits of the current speech synthesis techniques, see section 7.2) but it could be very expensive both in terms of time and money, so it is always a good rule of the thumb to look what we can get from speech data already available for speech synthesis.

Strom et al. (2006) and Strom et al. (2007) designed and built several speech corpora whose main aim is that of providing a phonetic coverage of the emphatic accents. One of these corpora, the "carrier sentences" corpus, consists of hundreds of emphatic words recorded in three different templates as in the following example:

S1: *It was JAMES who did it.*
S2 *No, it was JOHN who did it!*
S3 *It was JOHN, not JAMES*

The templates were repeated tens of times using different proper names. The speaker was asked to emphasize the names so the contrastive accents are not "spontaneous" contrastive accents and are often particularly strong.

The training speech data consists of all the 1683 utterances from the "carrier sentences" set and the 2025 "neutral style" utterances used to train the TTS systems of experiment 2. Compared to the training data used in Badino et al. (2009) the training data used in this experiment contains about 900 "neutral" utterances more as we found out that adding more neutral data helps smoothing the realization of the emphatic accents.

We built one new HMM based system (henceforth HTS05-PP-E) having the same linguistic feature set of system HTS05-PP of experiment 2 plus additional features dedicated to emphasis generation:

- {previous,current,next} phoneme emphasis value
- {previous,current,next} syllable emphasis value
- emphasis dependent name of the {previous,current,next} phoneme
- emphasis dependent name of the syllable nucleus

There are three possible emphasis values: 0 if the word is not emphatic, A if the word is the first or the only emphatic word in the utterance, and B if the word is the second emphatic word in the utterance. In the system described in Badino et al. (2009)

only two emphasis values were used, emphatic and not-emphatic. We increased the number of emphasis values as it emerged from an informal analysis that in the previous system the two contrastive accents were perceived as too similar and a differentiation seemed preferable.

Note that the test sentences only contain one contrastive pair each, so the first contrastive word has a A value while the the second has a B value.

The "emphasis dependent name of the {previous,current,next} phoneme" has different values on two identical phonemes if the emphasis values of the two phonemes are different. Again, as for pitch accents in experiment 2, we used this kind of feature to capture some possible specific phonetic effect of emphasis.

6.3.2 Test design

We designed two different listening tests. In one test, a preference test, we selected 20 sentences from the whole set of sentences where the *contrast* tagger correctly identified contrastive word pairs in the leave-one-out evaluation (see Appendix A for the whole set of sentence on which the *contrast* tagger was trained and tested). The selection criterion consisted in trying to have as many scenarios of *contrast* as possible in the test utterances. So, for example, we included *contrast* triggered by comparison (e.g., "**They** have probably had more time than **you** had to think about this subject") and by antonymy (e.g., "Every time we get a real **good** player they treat him **bad**).

Using system HTS05-PP-E we synthesized two different versions of the same sentence. In one version, contrastive words were accented with a standard pitch accent (version StdC) while in the other version the contrastive words were accented with an emphatic contrastive accent (version EmphC). The test participants were asked to indicate which version sounded best (the "no-preference" option was also available).

In addition to listening to the utterances the subjects had to read dialogue excerpts containing the test sentences. As already mentioned in chapter 5 the sentences on which the *contrast* tagger was trained and tested are sentences from the Switchboard corpus so that the dialogue excerpts are not fictional but excerpts from the Switchboard corpus (although some of them were slightly modified to avoid otherwise too long excerpts).

In the other test, an emphasis detection test, we selected 10 sentences containing at least one contrastive word pair and synthesized them with an emphatic contrastive accent on only one word (that could be a word of the contrastive pair if the sentence

contained more than one contrastive word pair). The remaining words were normally accented by the accent predictor. The subjects were asked to indicate the word they perceived as most prominent.

The presence of *contrast* (identifiable from text) in the test sentences had the aim of making the emphasis recognition task more difficult by giving to the listeners no textual cues or misleading textual cues about the placement of the emphatic accent. In fact the presence of *contrast* could lead the partipants to expect emphasis on the contrastive words while the emphatic accent was actually on none of the contrastive words or only on one of them (when the sentence contained more than one contrastive word pair).

The participants were exactly the same participants of experiment 2, they carried out the two tests soon after having carried out the two tests of experiment 2.

6.3.3 Results and discussion

EmphC	StdC	No preference	p-value_1	p-value_2
221	180	199	$p < 0.05$	$p = 0.094$

Table 6.5: *EmphC vs. StdC. In EmphC the contrastive words are marked with an emphatic contrastive accent while in StdC the same contrastive words are marked with a standard pitch accent. The first two columns show the number of preferences for one of the two versions and the third columns the number of cases in which subjects expressed no preference. The p-values are computed as in the previous experiments.*

The results of the preference test are shown in table 6.5. The preference for the emphatic accentuation of contrastive words is significant when removing the "No preferences" (with p-value < 0.05).

This results are much more positive than those achieved in Badino et al. (2009) although it is hard to say which factors mostly contributed to this improvement as several changes have been made to the training data (in the new system we added more "neutral" training data) and the feature set (e.g., in the new system there are three values of emphasis instead of just two (emphatic/non-emphatic)).

Comparing version EmphC and StdC at the sentence level, there is only one utterance pair where there is a significant preference for the EmphC version (both p-values

< 0.01). Concerning all other pairs there is no significant preference for either of the two systems: in 11 pairs there is a minimal preference for version EmphC, in 7 pairs there is a minimal preference for version StdC while in 1 pair the two versions have exactly the same number of preferences.

Looking at these results at the sentence level one might infer that in most of the cases the two versions EmphC and StdC are very similar because the emphatic contrastive accent is neither emphatic nor contrastive and is not distinguishable from the standard accent.

However this is not the case because the emphasis detection test shows that the emphatic accent is often clearly identifiable. In fact in 6 out of 10 utterances the number of speakers able to identify the emphatic word is significantly (with $p \ll 0.01$ in a binomial two-sided test) greater than the chance level (where the chance level is computed taking into account the emphatic word and all the accented words in the utterance).

In conclusion, results from both tests show that system HTS05-PP-E is the first HMM based TTS system able to identify contrastive words from text and to prosodically mark them with appropriate contrastive accents. An appropriate prosodic marking of contrast significantly increases the quality of the synthetic speech.

Nevertheless we believe that there is still a large margin of improvement in the generation of contrastive accents. For example, a quick improvement may be achieved by applying simple rules that replace the standard pitch accents once the contrastive accents have been placed and/or rules like "do not mark with an emphatic accent a contrastive word if that word is the last word in the utterance". Such a rule would have guaranteed more preferences for version EmphC in a couple of cases as the last accented word of a utterance is usually already perceived as the most prominent (i.e., it carries a *nuclear* accent) making an emphatic accent on it unnecessary and too strong.

6.4 Summary

In this chapter we have described three experiments in the generation of "standard" and contrastive accents in speech synthesis.

In the first experiment we have compared two unit selection systems, one having a target cost function that (indirectly) includes the information about the optionality of the pitch accent placements and one whose target cost function does not include such information. We hypothesized that the knowledge of the optionality of pitch accents allows to increase the number of acceptable candidate speech unit and so increases the

chances of selecting a good sequence of speech units. Results on a large scale listening test confirmed such hypothesis. We expect similar results when applying this approach to other prosodic events (e.g., prosodic breaks).

In the second experiment we have investigated the actual utility of accurate pitch accent prediction/detection in HMM based speech synthesis. Results from a large scale listening test show that a large improvement on accent prediction has no effect on the naturalness of the speech generated by the TTS system and so cast doubts on the actual utility of the $\pm$accent distinction for prosodic modeling in HMM-based speech synthesis.

In the last experiment we have described a method to generate contrastive accents in HMM based speech synthesis. We have then shown, through a large scale listening test, that (in HMM based TTS synthesis) marking contrastive words (i.e., words that result to be contrastive when looking at text only) with contrastive accents sounds better than marking contrastive words with "standard" pitch accents. This result stresses the necessity to go beyond a simple $\pm$accent distinction in order to improve the generation of prosodic prominence patterns in TTS synthesis.

Chapter 7

Discussion

This concluding chapter goes through the main contributions of this thesis, the problems encountered and not yet solved, and some discussions for future work in the identification and generation of prosodic prominence patterns in TTS synthesis.

Concerning the identification from text of accentuation patterns, the main contributions of this thesis are:

- an increased accuracy in ±accent prediction (chapter 3);
- a study on pitch accent optionality and its role on the evaluation of pitch accent prediction (chapter 4);
- a new approach for the identification of *contrast* (chapter 5);

On the generation side the main contributions are:

- an approach that integrates the optionality of prosodic symbols in the unit selection process in unit selection TTS synthesis (chapter 4 and 6);
- an investigation on the actual utility of pitch accent prediction in HMM based TTS synthesis that points out the limits of a prosodic model in which prosodic prominence patterns are "flattened" into sequences of ± accents (chapter 6);
- the generation of appropriate contrastive accents in HMM based TTS synthesis (chapter 6);

Like perhaps any research work this thesis has raised at least as many new questions as it has answered.

7.1 Pitch accent prediction

In chapter 3 we proposed a set of statistical and syntactic features that led to a 85.2% accent prediction accuracy in read speech and a 75.8% accuracy in conversational speech. After observing these results the question we posed was: how far is automatic accent prediction from perfect (i.e., manual) prediction?

When we take into account variability in pitch accent placement the answer can not be naively the difference between the 100% accuracy and our predictor accuracy.

From the evaluation of accent prediction presented in chapter 4 where the accent predictor was tested on multi-speaker data (i.e., test data in which six speakers read the same text) our accent predictor seems to have achieved a "perfect" accuracy, i.e.,

an accuracy that equals the accuracy of human prediction. An analysis of error, again carried out comparing the automatically predicted accents with accents labeled in the multi-speaker data, shows that there is very little margin left for improvement. The predictor only fails in few cases, mainly when function words are accented because they convey contrast, when the typical accentuation value of a word is inverted because of some lexical effects, and when content words are deaccented because the concept they convey is redundant.

Since the prediction evaluation and analysis of error on multi-speaker data has been carried out on the Boston University Radio News corpus the conclusion that pitch accent prediction has reached a perfect prediction applies to read speech in radio news style but may not apply to other speech styles such as spontaneous speech, "conference" speech, "tutorial" speech and so on. Radio news speech is over-accented (on average 50% of words are accented) with respect to other speech styles, and word accentuation is mainly due to the intrinsic informativeness of a word, while in other speech styles accentuation may be more dependent on context-dependent factors. For such reason it would be very useful having more prosodically labeled data in other speech styles. Multi-speaker data would be very useful to carry out the same prediction evaluation and analysis of error we carried out on the radio news style.

Achieving a perfect $\pm$accent prediction does not solve the problem of predicting prosodic prominence. Prosodic prominence is a relative property in that the prominence of a word is not absolute but relative to the prominence of the other words. As we have seen in section 2.1, according to the Autosegmental-Metrical Theory, the prominence pattern of an utterance is structured in a binary tree of weak-strong relations between words of which a $\pm$accent sequence is only a flattening, an approximation.

If parsing prominence trees seems very difficult to accomplish and/or of debatable utility in a TTS application (at least using the current speech synthesis techniques, see discussion in section 7.2), a probably more feasible and useful approach (for TTS purposes) consists in increasing the levels of prominence and so moving from the $\pm$accent distinction to the distinction between no-accents, primary accent (i.e., nuclear accent) and secondary (i.e., non-nuclear) accents in a prosodic phrase (and/or in a utterance), where a primary accent is the accent perceived as most prominent in a prosodic phrase.

An automatic discrimination between nuclear and non-nuclear accents has already been proposed by Calhoun (Calhoun (2006) and Calhoun (2008)) where part of the Switchboard corpus was annotated distinguishing between the three categories of accentuation "no-accent", "non-nuclear accent", "nuclear accent". Calhoun (2008) shows

that, according to our findings and previous findings, word accentability (i.e., the probability of a word of being accented with either a nuclear or a non-nuclear accent) mainly depends on the intrinsic informativeness of words while the probability of a word of bearing a nuclear accent (as opposed to a non-nuclear accent) mainly depends on its position in the prosodic phrase (which is expected because of the right-branching bias of the prosodic tree that makes sound the last accent in the phrase as the most prominent) and on whether the word is focused or not.

In Calhoun (2008) all context-dependent linguistic features, including *focus* (where *focus* is divided in the six categories mentioned in section 2.5), are manually extracted. In order to achieve a fully automatic ±nuclear accent classification (and then test its utility in TTS synthesis) we need to reliably identify *focus* first (and to have a good prosodic phrase predictor). The work on *contrast* tagging we presented in chapter 5 goes in this direction since *contrast* is (according to the semantic account on *focus* discussed in section 2.2.3.1) a special case of *focus*.

7.2 Pitch accents in TTS synthesis

In almost any study on automatic pitch accent prediction/detection TTS synthesis is mentioned as one of the applications that needs to model prosodic prominence and that would benefit from an accurate identification of accentual patterns. However the main concern of such studies is that of improving state-of-the-art accent prediction/detection accuracy and no questions are posed on the impact an improvement in accuracy would have on TTS synthesis (and/or on other applications).

We addressed this issue in experiment 2 in chapter 6 (section 6.2) where we tested the actual utility of accurate ±accent labeling in HMM based TTS synthesis. We compared a TTS system (system HTS05) that uses a poor-accuracy accent predictor with a TTS system (system HTS05-PP) that uses a state-of-the-art accent predictor. Both predictors are used to label both training data and test sentences. Results from a large-scale listening test do not show any statistically significant difference between the two systems.

Such result casts doubts on the actual utility of accurate pitch accent prediction, and, more in general, of ±accents, suggesting that, at least when synthesizing out-of-context test sentences, high-accuracy accent prediction is not needed and a simple function/content word distinction may be sufficient to account for accentual patterns in HMM based TTS synthesis.

The non-utility of pitch accents may point out the limits of an oversimplified prosodic prominence model which flattens the hierarchical prosodic prominence structure (i.e., the prominence tree) into sequences of ±accents and so can not properly capture the weak-strong prominence relations between items of an utterance. Making an analogy with syntax, the loss of information we get when we map the prosodic prominence structure into a sequence of pitch accents may be comparable to the loss of information (about the syntactic structure) we would have if we mapped syntactic dependency trees onto sequences of POS'.

However such conclusions, although significant, cannot be regarded as conclusive yet. One of the reason is that the accent labeling of the training data is not a prediction task but a detection task and, and the accent detection accuracy might be far from the "perfect",i.e., human-like, accuracy, as opposed to the accuracy of our accent predictor (see section 4.4 and discussion at the end of section 6.1.3).

Additionally, since our predictor was tested (and trained) on a speaker different from the one used to train the HMM based system, we do not know the detection accuracy on the TTS system's training speech but we have reasons to believe it is not close to the "perfect" detection accuracy and we guess (testing the predictor on different voices of the Boston University Radio New corpus) it is around 75%-85%.

We might have integrated acoustic features in our predictor (transforming it into an accent detector) but as we have seen in section 2.4 we expect the benefit arising from the use of acoustic features to be very small. An effective but time consuming solution would be that of manually labelling the training data, while other perhaps less effective but also less time consuming solutions would require speaker adaptation techniques. Following up the considerations made in the previous section and considering (so far) the uselessness of ±accent, a future direction in the generation of prosodic prominence could be that of going beyond a binary categorization of prominence, for example by differentiating between nuclear and non-nuclear accents and/or by including in the training feature set of the TTS system features that incorporate strong-weak prominence relations (even between two accented words).

Actually in system HTS05-PP the predictive features of the accent predictor were added to the TTS system training feature set with the aim of capturing weak-strong prominence relations between adjacent words that can not be captured by a gross ±accent distinction. Moreover, since the placement of a nuclear accent is strongly affected by prosodic structure (as it is often the last accent in the prosodic phrase) which in turn is strongly constrained by the syntactic structure, the use of the syntactic

features (extracted from a dependency parser) was also intended to indirectly incorporate a distinction between nuclear and non-nuclear accents. These features failed to bring any improvement, but contrary to the accent feature they were only present at the "word" level (i.e., the features referred to word properties and not syllable or phone properties). It would be of interest (and of easy implementation) to take such features at the phone level and test again their utility.

Suppose we have an accurate prediction and detection of nuclear accents, our expectation is that the ±nuclear discrimination will be particularly important when the utterance to synthesize is in context and the context, through *focus* marking, will force the nuclear accent to be in a position different from the "default" position (i.e., the position that the nuclear accent would have if the utterance were "out-of-context").

Nevertheless having a perfect identification of the nuclear accent may not be sufficient to make sound the nuclear accents as the most prominent. The fact that nuclear accentuation is a relative property determined by weak-strong relations between words (i.e., an accent is nuclear because it is perceived as stronger than all other accents in the phrase and not, or at least not necessarily, because it has some distinctive phonetic properties) might be problematic with the two current state-of-the-art speech synthesis techniques. In general any prosodic category that is relative and has no clear distinctive phonetic correlates may not be modeled in unit selection and HMM-based speech synthesis.

Consider the example:

(1) Q: Who went to Madison Square Garden?
A: My mother-in-law went to Madison Square Garden.

in which (if the utterance is a single whole prosodic phrase) the accent on "mother" has to be the most prominent accent (more than the accent on "Garden", which would be the nuclear accent in the "out-of-context" case).

In HMM based speech synthesis the strong-weak relation between "mother" and "Garden" might not be reproduced as the phonetic realization of an accent depends on a restricted linguistic context that cannot represents relations between far apart words. Figure 7.1 (top) shows a standard HMM where the probability of an acoustic vector at time t only depends on the hidden state occurring at time t. Suppose that the hidden state at time t is the stressed syllable of word "Garden". If nuclear accents do not have distinctive phonetic correlates then the linguistic information contained (and which defines) the hidden state at time t cannot guarantee that the generated acoustic vector at time t will result in a less prosodically prominent syllable than the stressed syllable

of word “mother”.

In order to generate this relative prominence relation more information at time t is needed: we need to know how the stressed syllable of “mother” was prosodically realized, i.e., we need to know the values of the hidden and the observation state associate to it. That is possible by adding new statistical dependencies between an observation node and a fixed number of previous hidden and observation states as it is shown at the bottom of figure 7.1. Such a model presents several problems, the most severe of which is that a much larger number of statistical dependencies implies a much larger number of statistical parameters to be learnt, which in turns implies a massive problem of data sparsity.

A similar reasoning applies to unit-selection.

For the same reasons, parsing prosodic prominence trees does not seem to guarantee a good generation of prosodic prominence. We could map any parsed prominence tree into a prominence grid (see bottom of figure 2.1 in chapter 2) where the degree of prominence of any syllable is specified. We could then use the degree of prominence as one of the linguistic features of our TTS system.

However if these degrees of prominence do not have clear distinctive phonetic properties then we end up with the same problem described above.

Because of the limit of the current speech synthesis techniques we may still improve the generation of prosodic prominence by applying some approximate solutions, i.e. solutions that make strong assumption about the prosodic prominence model. For example an approximate solution for cases in which the primary accent is not in the default position as in example (1) consists in: 1) assuming that primary accents are stronger than any other type of accent (which implies that they have clear phonetic correlates); 2) including in the training data (or in the speech database) particularly prominent accents that would sound as the most prominent in any context; and 3) learning (or “selecting”) those accents to generate primary accents. We have successfully applied this solution in the TTS system described in section 6.3 in order to generate contrastive accents.

7.3 *Contrast* labeling

Chapter 5 addressed the problem of the automatic identification of *contrast* where *contrast* is a particular scenario of focus in which the focused word is a word that explicitly contrast with another.

Working on the identification of *contrast* stems from the necessity of going beyond the ±accent distinction and taking into account the effect of focus on prosodic prominence.

The small number of (positive) examples of *contrast* extracted at each conversation and used to train and test our *contrast* tagger might lead to think of *contrast* as a very rare event and to doubt of its actual utility in TTS synthesis. However this is not the case as the conversations contain many more examples of *contrast* (not only on words but also on phrases) and the limited number of examples we used is due to the need of tackling a complex and novel problem starting from a simplification of it. In fact in order to simplify the task we had to remove all examples of *contrast* spanning over two sentences or more and all the examples of *contrast* on syntactic phrases (instead of on single words).

As discussed in section 5.3.1, extending the identification of *contrast* to *contrast* on phrases, not only would drastically increase the number of examples of *contrast* but could also improve the accuracy in the identification of *contrast* on words. The identification of contrastive syntactic phrases could be addressed again by first simplifying the problem, for example by only trying to identify "short" noun phrases. Obviously the task would require the identification of all noun phrases first.

Whether the *contrast* is on words or phrases we believe that the identification of *contrast* intended as a pragmatic relation identifiable from text should be neatly separated from the prediction of its prosodic realization, in other words *contrast* identifiable from text and contrastive accents on contrastive items should be kept as two distinct concepts. Such separation was not entirely applied in our approach to *contrast* identification, since only prosodically marked *contrast* was annotated in the corpus we used, and that hampered the accuracy of our *contrast* tagger.

The separation of the two concepts would first allow to achieve a better identification of *contrast* and then to identify the factors that make an instance of *contrast* most prone to a strong prosodic marking than other instances. Among these factors, "subjective language" (i.e., language denoting subjectivity, see Wiebe et al. (2004)) seems to be an important one. Consider the two sentences:

(a) **John** was doing all the **cooking** and **Kate** was doing all the **enjoying**

(b) **I** was unfairly doing all the **cooking** and **Kate** was doing all the **enjoying**

In both sentences there are two contrastive word pairs but in sentence (b) the probability that the contrastive words will be marked by contrastive (or emphatic) accents

seems to be much higher than in (a) because of the presence in (b) of the "subjective" items "I" (in place of John) and "unfairly". The presence of items from subjective language denotes a personal involvement of the speaker that we expect to be (usually) prosodically marked with increased emphasis (not necessarily only on the contrastive words).

The separation of *contrast* from its prosodic realization offers some advantages but also poses a problem that is mainly due to its loose definition, which, lacking of an algorithmic nature, leaves space to several ambiguous cases. In fact defining (symmetric) *contrast* as the relation linking to words that explicitly contrast with each other , does not explain what *contrasting* means. In the sentence:

(c) John went to London but Kate went to Paris.

the *contrast* between "John" and "Kate", and "London" and "Paris" is evident, while in:

(d) John went to London and Kate went to Paris.

contrast seems less evident since the two clauses "John went to London" and "Kate went to Paris" are linked by a sequential discourse relation instead of a contrast discourse relation.

However if, according to the semantic account on contrast/focus (see section 2.2.3.1), we define two contrastive words as words that evoke each other and agree that the "evoking" is activated by the symmetry of the two clauses 'John went to London" and "Kate went to Paris", then "John-Kate" and "London-Paris" have to be considered contrastive. Yet, in a sentence containing symmetric clauses like the following:

(e) John had a sandwich and then he had a salad.

we can hardly see any cue of *contrast* between "sandwich" and "salad" despite of the symmetry between the two clauses.

On the other hand if we had access to a prosodic realization of (d) in which "John", "Kate", "London" and "Paris" carry a contrastive accent than we could affirm without doubt that the four words are contrastive as the "evoking" is triggered by the prosodic signal (i.e., the strong accent on "Paris" would imply that is "Paris" where "Kate" went and not "London").

Perhaps examples (c), (d) and (e) all contain *contrast* but at different degrees. What we need is a formal definition of *contrast* that states the procedure to manually label *contrast* (from text only).

7.4 Generation of contrastive accents

In the experiment of section 6.3 we described an approach to generating contrastive accents in HMM based TTS system and showed that when the TTS system prosodically marks contrastive words (i.e., words that are contrastive on "textual" basis) with contrastive accents, the naturalness of the synthetic utterances increases with respect to utterances where contrastive words bear "standard" accents.

In order to generate contrastive accents we trained the TTS system on data containing emphatic contrastive data (i.e., it-cleft utterances where the speaker was required to emphasize the contrastive words) and neutral data. From an (informal) comparison with some previous attempts we made, it emerged that a balanced trade-off between the two types of training data is crucial for a successful generation of contrastive accents. Given a fixed amount of contrastive speech, adding neutral speech serves two purposes: 1) it smoothes the excessive prominence of the emphatic (contrastive) accents, 2) it improves the general quality of the synthetic speech. However an excess of neutral data may cause the contrastive accents to lose their peculiar prominence and conflate into standard accents.

Further improvements in the generation of contrastive accents might be made by searching for a better trade-off of the two data types (in fact there is no guarantee that the ratio we chose is optimal) and/or by adding new training data containing several scenarios of *contrast*. The emphatic speech we used consists of hundreds of utterances presented in only three templates, of which one is a scenario of symmetric *contrast*. A training set containing many more scenarios of *contrast* may allow to better model the prosody of the contrastive words and of their neighbour words. Such a training set could be built by using our *contrast* tagger to collect sentences containing *contrast*.

In the hypothesis that contrastive accents only differ from standard accents in that they are simply perceived as more prominent we may use contrastive accents to generate nuclear accents (whether the word carrying the nuclear accent be contrastive or not), especially when the last (predicted) accent is not the last accent in the prosodic phrase.

We have carried out some preliminary experiments (see Andersson et al. (2009)) in which all focused words in automatically generated answers (to questions from users of a dialogue system for restaurant booking) were accented with a contrastive (emphatic) accent. The accents were often perceived as too strong as we used our first TTS system for *contrast* generation (Badino et al. (2009)) but the emphasis on focused words led to

a largely increased prosodic appropriateness with respect to utterances where focused words were not distinguished from all other words. We expect that using our last TTS system (i.e., the one described in section 6.3) to mark the focused words will further increase appropriateness, intelligibility and naturalness.

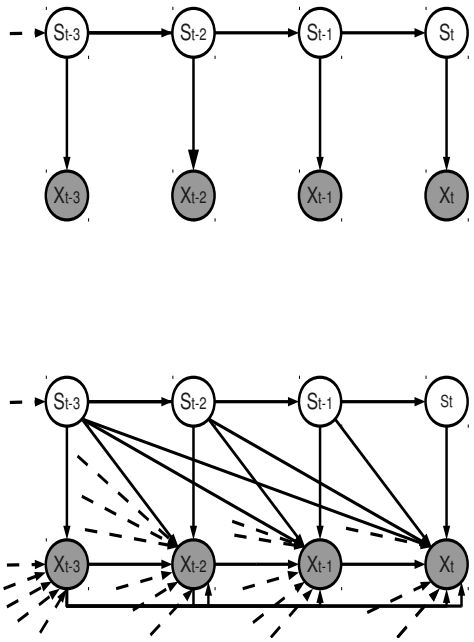

Figure 7.1: *From 1st order HMMs to "fully connected" HMMs. The shaded nodes represent the observation nodes, i.e., the vectors of acoustic parameters, while the clear nodes are the hidden nodes, i.e., nodes defined by the value of the basic speech unit (e.g., phoneme, triphone, syllable, etc...) plus linguistic properties of that value. The top figure depicts a standard HMM where the probability of an acoustic vector at time t only depends on the hidden state occurring at time t. On the bottom a "fully connected" HMM is shown, where the probability of an acoustic vector at time t not only depends on the hidden state at time t, but also on the previous k hidden states and on the previous k observation states.*

Appendix A

The following is the list of all example of *contrast* used to train and test the fully automatic *contrast* tagger of chapter 5 (section 5.3). The capitalized words are the contrastive words.

1) there 's, , things that invade that second type of privacy where you do KNOW about them and possibly things that invade that second type of privacy without you KNOWING about it,

2) I finally just got to where I go, okay, **I** 'm Debbie Moore, may I ask who YOU are and what you are in my classroom for,

3) anything that Tom Landry WANTED or had, he was going to CHANGE it.

4) every time we 'd get a real GOOD player they 'd treat him BAD.

5) they had to kick. to know whether we were going to WIN the game or LOSE a game because they got one point ahead of us.

6) There 's no question about it, because EVERYBODY was playing good together except the QUARTERBACKS.

7) The one that was in for the SEVENTEEN years, actually served SEVEN,

8) the REASONS that they 're doing it compared to the REASON someone else is in jail for it, it 's like two different things.

9) you take Asian countries, or, the Eastern countries where WOMEN are in the background and the MEN are in the foreground,

10) you take Asian countries, or, the Eastern countries where women are in the BACKGROUND and the men are in the FOREGROUND,

11) it 's like, THEY live back in where WE came from two hundred years ago.

12) I 'm not saying that they 're all good either, because there 's GOOD and BAD in everything,

13) he TRADED the other one in and GOT this one,

14) you see WOMEN going off to wars as well as MEN.

15) after that, if you asked me that I would n't have been able to tell you if that was FOOTBALL or BASEBALL.

16) when it HAPPENED he MISSED it, and did n't understand it

17) he was sort of in between STRASBOURG, and PARIS.

18) I could go along with that, if I could be assured that it would be their natural life in JAIL and not PAROLE after ten, or twelve years.

19) the difficulty would be in whether it 's VOLUNTARY or INVOLUNTARY.

20) there has to be some sort of punishment. Short of physically ISOLATING the child, and short of physically HITTING the child.

21) they 'd much rather go in the store and BUY something than be SPANKED.

22) they 're definitely going to WORK towards being good, rather than trying to ACT up and be bad,

23) they 're definitely going to work towards being GOOD, rather than trying to act up and be BAD,

24) over in the Mid East, especially Israel, it 's just like ISRAELIS have a totalitarian system, when it comes to the PALESTINIANS.

25) HE knew what was going to happen, more than any of US.

26) I guess we 'll have to see another generation to see what differences a child being brought up, in kind of a, COMMUNITY, rather than a HOME.

27) I guess I see that as not so much a DEMAND but a PRIVILEGE.

28) I guess, I do n't see, this being that DIFFERENT. But even more BENEFICIAL because it would be something that everybody participated in, and would take a turn in.

29) that 's why THEY 're there and not ME.

30) those who DO n't vote WOULD have voted exactly the same way, in other words if forty percent had voted for that person and sixty percent for the other, just like everyone who did vote, it 's not clear to me that it is really a problem,

31) those who do n't vote would have voted exactly the same way, in other words if FORTY percent had voted for that person and SIXTY percent for the other, just like everyone who did vote, it 's not clear to me that it is really a problem,

32) Anyone who does n't vote, it 's fine with me as long as I can have THEIR vote.

33) I sort of feel that way. especially when, they do n't vote for someone because they do n't like any of them and then the person gets in and they do n't like him and HE turns out to have been worse than HER that they might have voted for, or something like that.

34) I think it was Hamilton who wrote number ten or something, where he was arguing for a REPUBLICAN, not in the sense of the Republican party now, versus a DEMOCRATIC, government and, arguing successfully, why, the United States should be a republic, not a democracy. Which indeed it really is, a republic, not a democracy.

35) I think it was Hamilton who wrote number ten or something, where he was arguing for a republican, not in the sense of the Republican party now, versus a democratic, government and, arguing successfully, why, the United States should be a REPUBLIC, not a DEMOCRACY. Which indeed it really is, a republic, not a democracy.

36) I think it was Hamilton who wrote number ten or something, where he was arguing for a republican, not in the sense of the Republican party now, versus a democratic, government and, arguing successfully, why, the United States should be a republic, not a democracy. Which indeed it really is, a REPUBLIC, not a DEMOCRACY.

37) Where he defines DEMOCRACY as everyone votes for the issues and REPUBLIC is people who vote for someone who then in turn votes for the issues.

38) Where he defines democracy as everyone votes for the ISSUES and republic is people who vote for SOMEONE who then in turn votes for the issues.

39) if there 's some MINORITY that people do n't like because of, racial hatred or something like that, the MAJORITY can just simply vote against them.

40) I think they 're looking after their own self PRESERVATION more than they 're actually looking after the GOOD of the country.

41) I understand the void that, comes naturally with both RADIO presentations and TELEVISION presentations

42) on Friday most everybody wears jeans and SWEATSHIRTS, or, jeans and BLOUSES or something like that.

43) he did n't carry any ties or anything because he just went over there with that attitude, if I do n't TAKE it then they wo n't make me WEAR it.

44) I 'm convinced that, Detroit or whoever it is made a major mistake, years ago when they stopped putting the small V EIGHTS in and went to the FOURS and some of the sixes.

45) I 'm convinced that, Detroit or whoever it is made a major mistake, years ago when they stopped putting the small V EIGHTS in and went to the fours and some of the SIXES.

46) looking BACK, maybe some of the things that I know NOW, I 'm not sure I do believe it was worth the cost in dollars and lives.

47) I feel like maybe they felt like we were doing the right thing to try and help

maintain the DEMOCRACY over there and beat the COMMUNISM,

48) nobody says because you 're supporting the TROOPS that you 're supporting the WAR.

49) if you do n't WANT the kids, then it 's not the time to HAVE,

50) I think it would be neat if they could incorporate into small and large businesses both a built in day cares where the children were THERE in the facility but not necessarily RIGHT there with you.

51) I favor DOGS over CATS actually

52) if you get them YOUNG and everything before they go, kind of NUTS,

53) like you said, save the COMMENTARIES because I 'm going to listen to the NEWS and draw my own opinions.

54) the ground will filter SOME of it but not ALL of it.

55) for what 's cut down HERE, more will be cut down someplace ELSE

56) they do because, what is it, carbon dioxide, THEY use that where WE ca n't synthesize it

57) they do because, what is it, carbon dioxide, they USE that where we ca n't SYNTHESIZE it

58) we could talk a bit about, just quality of products in general, if they 're BETTER or WORSE like if they last longer or something like that.

59) the problem is that you CA n't put the type of card that we need to put in it, in it. even though they told us in advance that you COULD,

60) WE do n't want YOU to go through the work and then find out that this does n't really work anyway, and et cetera,

61) second thing, we still CA n't put the boards in, because they have other things blocking where we need to DO it.

62) I just tell the credit card company, do n't PAY the charge, even though they 've already PAID it.

63) they 'll just, UNDO the credit that they DID to them.

64) THEY will put the kind of clout on them that YOU ca n't really do.

65) pretty much spent most of my time either in the YARD or at NURSERIES buying stuff for the yard.

66) cars I think are not MADE, as well as they COULD be.

67) WE do n't really care about YOU, as an individual.

68) I do n't DO it as much as I SHOULD.

69) I 'm not DOING it as much as I NEED to,

70) I think another thing is that my friend that was TAKING with me the first two months WAS n't the second two months.

71) I think another thing is that my friend that was taking with me the FIRST two months was n't the SECOND two months.

72) I just do n't TAKE as much advantage of it as I SHOULD.

73) I guess, they tried what 'd call it, SOFT conversion and HARD conversion

74) I think they just go all the way on new products introduced or whatever, start your packaging, go to LITERS instead of QUARTS,

75) the differences between SEVENTY and SEVENTY-TWO or seventy-five degrees is n't much,

76) the differences between SEVENTY and seventy-two or SEVENTY-FIVE degrees is n't much,

77) the difference THIRTY and THIRTY-FIVE degrees is, quite a bit.

78) It was made to placate some of the NORTHERN support but not completely alienate all the SOUTHERN support because, if you read it, it only emancipated those who were in areas, in rebellion against the United States.

79) until we HAVE to learn to think that way, we WO n't.

80) we have to separate our PAPERS, and our GLASS,

81) I noticed at the library or someplace this past, month, month and a half ago, they were having a speaker, talk about doing lawn work and, how important it is to cut your lawn without a bag. Just to kind of MULCH it, rather than BAG it up, because of all the grass that 's being bagged and being hauled away by the garbage trucks and stuff,

82) they do n't EXPAND or CONTRACT when the weather changes,

83) when I READ it, I should have REALIZED that because, it had Foster 's style written all over it.

84) sometimes the more you GET the more you WANT too,

85) it takes me almost the same amount of time to go to Frederick, as it does to go to the ballpark in, BALTIMORE. Because, to go out to FREDERICK, I just jump on the Interstate

86) put them to WORK rather than SITTING there.

87) that again would CUT some of our budget down for education but BUILD up our education with the people. at high school level, which I would like to see. Rather than so many kids getting out of school.

88) the MONEY being spent and the GOODS flow in and we do n't, sell products

abroad as much as we bring in.

89) the money being SPENT and the goods FLOW in and we do n't, sell products abroad as much as we bring in.

90) I got about a HUNDRED pages through it and realized I had a THOUSAND more,

91) one of the news stations. they had on there, where the output from the United States was basically from SMALLER businesses rather than the LARGER ones are exporting.

92) they were n't supposed to PROCESS it until they DELIVERED it.

93) they 're CONVICTING them faster than they 're EXECUTING them

94) of course, they 're playing it up real BIG that it looked like it was a very, very SMALL ring

95) we are there by going to remove them from society, not TEMPORARILY, but PERMANENTLY.

96) also, the thing I do n't like about a lot of these court trials and a lot of these appeals is that it 's not based on what is TRUE and what is FALSE

97) rather it 's on the rules of evidence, what can I HIDE and what do I have to TELL.

98) when the guy came out with the bag they thought was MONEY, it turned out to be a couple jumbo COOKIES

99) they 've, decide the PENALTY separately from the VERDICT.

100) these are the kind of people I would like to get out of here and get out of circulation and say we do n't accept YOUR kind in OUR society.

101) if the parents are n't SUPPLYING it, they 've got to GET it from someone else from the schools,

102) I felt probably worse for THEM than for ME

103) gosh this fellow 's wife who was, watching the dog, I think SHE loved that dog more than I did

104) this dog, Dennis jumped in and got this look on his face like what do I do now, as he 's FLOATING down the river then finally discovered that he could SWIM and, actually paddled up against the current and, made it back to the shore and climbed up

105) every time I WANTED one I have to go outside and just TAKE one. One at a time.

106) then YOU must know a lot more about this than **I** do.

107) the problem is I ca n't guarantee that a JUDGE would necessarily be much better than a JURY,

108) they pay those JURY members very little money compared to that JUDGE.

109) I PAY a good deal of taxes I guess, because I MAKE a fair amount of money,

110) I pay a good deal of TAXES I guess, because I make a fair amount of MONEY,

111) now, even the money that been RAISED for the local districts is going to be SIPHONED off and sent to other parts of the state.

112) They went from ELEVEN to FIFTEEN hundred,

113) the parents might have hostilities towards them, Like YOU 're judging US,

114) **I** grew up about the same time YOU did,

115) Do youself have children who ARE or have BEEN through the public school system ?

116) I would use the money that we 've paying them to, provide some special help in training and particularly, mentor teachers to work with the beginning teachers and the teachers who may have been at it a long time but have been making the SAME mistakes for a LONG time.

117) more you see these commercials that have Jimmy walking into class late and it happens that the TEACHER is an instructor who is in New York while JIMMY 's in Rome.

118) more you see these commercials that have Jimmy walking into class late and it happens that the teacher is an instructor who is in New YORK while Jimmy 's in ROME.

119) there 's a difference too, between EAST and WEST, in the south. the way in which people speak.

120) Where you 've got to LEAVE the furniture just one certain way, you ca n't REARRANGE it at all.

121) it runs about ten minutes FAST, except for about a month the clock ran NORMAL.

122) we completely stripped all the old WOOD shingles off, put DECKING up, put the paper down, and just started from scratch.

123) crimes against PROPERTY seem to outnumber crimes against LIFE,

124) Do you live in a real small TOWN or out in the COUNTRY ?

125) the other side of that might be if someone FOUND out something or SURMISED something that were n't true then I would feel probably more invaded in the gossipy sort of sense.

126) I really agree with the, INNOCENT until proved GUILTY theory.

127) I think what they need to do is STOP building more jails and START giving stiffer sentences,

128) I think what they need to do is stop building more JAILS and start giving stiffer SENTENCES,

129) you ca n't expect in a CLASSROOM for a particular course an hour a day to counteract, sixteen or seventeen years of influence at HOME.

130) I remember I had a college professor who once said that genius is one percent INSPIRATION, and ninety-nine percent PERSPIRATION.

131) We usually watch the LOCAL news and the, NATIONAL news both.

132) there 's a big difference between BALTIMORE and WASHINGTON, even though they 're so close.

133) WASHINGTON, it really is an international city, where BALTIMORE is hometown Baltimore.

134) I 'm more accustomed to a ONE acre lot being a standard and the TWO acre being what most people have.

135) it seemed like the FURTHER I got away from Dallas, center, the more lot came with the house and the LOWER the price, at the same time.

136) you know what they 've started doing is instead of the tail pipes being at the BOTTOM of those buses they 've started putting them up at the TOP,

137) **I** have probably had more time than YOU have to think about this subject,

138) during that week you 're kind of an ad hoc, group of, I do n't know, TWENTY-FIVE instead of TWELVE

139) the public would rather hear something NEGATIVE about the other guy than a POSITIVE factor.

140) I feel like,, if we did that people would have a lot higher confidence that their VOTE was counting rather than their CONTRIBUTIONS would count.

141) I think, you still have a view that the AMERICAN voter is different from OTHER voters.

142) I think that the JUDGES should be left to do most of the sentencing, simply because, there is always a JURY that might be swayed, by the moment, either to be too lenient or too vengeful, I guess.

143) my impression of it is that it has DOUBLED in the last ten years and TRIPLED in the last twenty.

144) I have got some in the BACKYARD that bloomed blue. Which I would have

liked those in the FRONT because they match my porch and stuff better. And then some on the side of the house with the dusty purple color. With little purple spots that it will fade into a solid purple.

145) I have got some in the BACKYARD that bloomed blue. Which I would have liked those in the front because they match my PORCH and stuff better. And then some on the side of the house with the dusty purple color. With little purple spots that it will fade into a solid purple.

146) I just threw them on the side, intending to TRANSPLANT them or throw them away or something. And FORGOT about them through the whole Winter.

147) I think sometimes it is just, HE and I are very different in terms of that.

148) most people talk about the NOISE pollution from airplanes rather than the AIR pollution.

149) I was reading the other day not to go on with this but that, DIESEL fumes actually have less pollutants in them than GASOLINE fumes.

150) if you want to, your LAWYER or your OPPONENT need to go face this group of twenty-five or a judge like they have on T V

151) I guess you 're better off sitting behind a BUS than a CAR although I could never, really rationalize that while I was sitting there.

152) we just feel that when we leave, this area, we 're going NORTH, not SOUTH.

153) I did a Sunday school lesson one time on the difference between the OLD Testament and the NEW Testament where there 's a vengeful Lord in the Old Testament and there 's a loving Lord in the New Testament.

154) I did a Sunday school lesson one time on the difference between the Old Testament and the New Testament where there 's a VENGEFUL Lord in the Old Testament and there 's a LOVING Lord in the New Testament.

155) I did a Sunday school lesson one time on the difference between the Old Testament and the New Testament where there 's a vengeful Lord in the OLD Testament and there 's a loving Lord in the NEW Testament.

156) is it doing a GOOD job or a BAD job

157) even a BAD school is a GOOD school up here, where, if I lived in New York City or Washington, D C, I would seriously consider moving if I had a child.

158) having been through grade school up THERE and coming down HERE to high school I can understand why.

159) even the PUBLIC schools are behind the PAROCHIAL schools.

160) there 's too many KIDS and not enough TEACHERS

161) if the teachers are getting six percent raises every YEAR when people in industry have been getting cut back, and you 're getting raises every eighteen MONTHS you got to go now, hey, wait a minute,

162) I would feel not only invaded in the sense that someone had OBTAINED information that I would rather they DID n't

163) Course on merchandise that I was buying on CARDS, I was getting the MONEY back,

164) BACK when I was going to school you just did n't get away with the things these kids get away with NOW.

165) I always tried to UNDERSTAND things, not tried to MEMORIZE.

166) they 're throwing more money at it NOW than ever BEFORE

167) the other part of it is PARENTS have quit becoming PARENTS.

168) they try to ENCOURAGE you to follow a specific curriculum, although you do n't HAVE to.

169) YOU take this subject much more personally than **I** do, I suppose.

170) for example if you 'd been a TECHNICIAN instead of an ENGINEER.

171) if it 's not MOM then DAD or somebody got to move in there and do the job, because the kids really need it.

172) if it 's not MOM then dad or SOMEBODY got to move in there and do the job, because the kids really need it.

173) if I had been allowed to work, maybe THIRTY hours a week instead of FIFTY hours a week, I might still be working basically full time or part time, if there had been some way to work it out,

174) people walk IN OUT

175) as far as housework goes, MEN can do housework just as easily as WOMEN,

176) the boys in this next generation are not going to have to be told as much this needs to be done, because MOM was there saying that DAD is there,

177) They 're going to say to the kids you need do this, because it needs to be done not because it 's a woman 's job or a man 's job, but because it 's DIRTY and it needs to be CLEAN.

178) I felt like when they were young, that was the time to instill it. That it could be added to and strengthened as they grew OLDER, but when they were LITTLE,

179) if you teach them when they 're LITTLE the way you want them to be, and the things that are important to you, then you add onto it as they get OLDER.

180) it could be a flame stitch where rather than DRAWING a picture you ’re MAKING a design like a geometric or whatever,

181) I STARTED a project, that I swore I was going to FINISH for somebody for Christmas.

182) That ’s pretty interesting just because you KNOW a subject matter does n’t mean you can TEACH it.

183) they ’ve quoted statistics that my THROAT just about fell into my TOES.

184) if they ca n’t, if they have missed that training, then somebody, before you ’re start PENALIZING them with bad grades for not being able to communicate what they ’re thinking, TEACH them these basic skills.

185) as a WESTERNER in INDIA, I was often surprised, and felt my sense of privacy there was quite invaded.

186) seeing what drugs DID to him makes me realize what it COULD do to people, in the work force as well.

187) YOU do a lot more area than **I** do.

188) I DID something a little bit different this year that I have n’t DONE before,

189) I think it would probably depend on whether this was a FORMAL or a IN-FORMAL dinner party.

190) also to get totally off the subject of CRAWFISH lots of VEGETABLES and hors d’oeuvres and stuff like that for a dinner party I think that that really helps

191) also to get totally off the subject of CRAWFISH lots of vegetables and HORS d’oeuvres and stuff like that for a dinner party I think that that really helps

192) before I had my second child we sort of had a contest going where HE would cook and then the next time it would be MY turn and I ’d try to outdo him and then he ’d try to outdo me

193) before I had my second child we sort of had a contest going where he would cook and then the next time it would be my turn and I ’d try to outdo him and then HE ’d try to outdo me

194) before I had my second child we sort of had a contest going where he would cook and then the next time it would be my turn and I ’d try to outdo HIM and then he ’d try to outdo ME

195) we threatened to make the other two make us dinner one time Just to even it out since WE seemed to be unfairly doing all the cooking THEY were doing all the enjoying

196) we threatened to make the other two make us dinner one time Just to even it

out since we seemed to be unfairly doing all the COOKING they were doing all the ENJOYING

197) A SHOTGUN hurts worse than a PISTOL does.

198) I think you could recover from a PISTOL but not from a SHOTGUN.

199) We traded his PISTOL for that SHOTGUN.

200) they start asking questions and in the sense are INVADING your privacy although, if you know what the social norms are, quote unquote, you ASKED for it.

201) I 've never really joined a club because I have n't got the TIME. Not because I have n't got the DESIRE.

202) Do you believe there ought to be legislation guiding the, BUYER and the SELLER ?

203) the PERSON who sells the gun ought to protect themselves because if that gun 's registered to them and SOMEBODY else uses that gun in something, the cops are going to come to you.

204) is n't there a way to DEREGISTER yourself after you REGISTER a gun ?

205) I kind of worry about getting a car that 's that new with low mileage on it because you wonder why did the person that OWNED it want to get RID of it.

206) she 's going to SELL it rather than TRADE it in.

207) after WASTING the first six years, partying and everything else, I decided, uh-huh, time to settle down and DO something.

208) Course my job was such that I could n't DO it as much as I WANTED

209) you think about the layer of bureaucracy between the MONEY and the RECIPIENT,

210) they 'll help the country eventually, too, because rather than having a bunch of UNEDUCATED people we can have EDUCATED people,

211) they use ELECTRONIC and ACOUSTIC interchangeably,

212) PARENTS are n't PARENTS.

213) So many times, you had the COAL miners and STEEL workers going out at the same time.

214) So many times, you had the coal MINERS and steel WORKERS going out at the same time.

215) when I was in JUNIOR high and HIGH school that never happened.

216) the woman who OWNED it SOLD it.

217) THEIR attitude and philosophy was just completely, opposite from MINE,

218) in fact I 'm doing that right now with their afternoons, when THEY get home before I do.

219) HE has enough relatives to make up for ME not having any.

220) HE had to kind of get used to US when we first started going together.

221) it has everybody that you COULD ever imagine, and some you COULD n't.

222) maybe those are not the clothes that are the most, APPEALING to you, or the most, COMPLIMENTARY to you.

223) someone else is telling me, okay, let 's move THIS way, let 's move THAT way, instead of me having to think about it so much.

224) I think, of course, now I go the other extreme, I do not like to see, in the corporate areas, all the WOMEN dressed like MEN, with the suits and, white shirts and ties and what have you so that they all look exactly the same.

225) you could hardly tell the WOMEN from the MEN except for the lengths of the pants. one was a skirt and one was a pant.

226) gosh, we 'RE keeping these men in prison for fifteen or twenty years on death row, and not DOING anything with them,

227) like I say, I do n't think the guy who 's going to rob a Seven Eleven, is going to rob a Seven Eleven whether he has a GUN or a KNIFE, baseball bat, or, whatever,

228) there are a lot of kids who when they 're TEN look like they 're TWELVE, or fourteen and, especially some of the minority children. Whether, a racist or not,

229) there are a lot of kids who when they 're TEN look like they 're twelve, or FOURTEEN and, especially some of the minority children. Whether, a racist or not,

230) if some of them are DOING something that they SHOULD n't be and then they have one of those toy guns in their hands

231) They can play with ANYTHING and make it a GUN.

232) they 'd have that respect for what it DOES, and what it CAN do.

233) boy if someone did break in, I do n't know that I COULD or WOULD even grab it.

234) It was like, YOU come after ME, you 're going to get it.

235) one of the things that 's in their culture that I really think the major corporations should pay attention to, is the fact that, while Japan was becoming a great power, financially, the people that worked for those companies, worked for the same company they worked for at SIXTY-FIVE, as they did when they were EIGHTEEN.

236) I should probably go back and read the book NOW that I just saw the movie again not too long AGO.

237) I was afraid that RAMBO was going to do the same thing that ROCKY was going to do, Go into fourteen hundred episodes.

238) I ca n't seem to communicate with a CAT like I can with my DOG.

239) I think, in a way we 're able to read each other pretty well because, SHE knows when **I** 'm upset and I know when she 's not feeling good too.

240) I think, in a way we 're able to read each other pretty well because, she knows when I 'm upset and *I* know when SHE 's not feeling good too.

241) Now some people object during primaries, having to declare a party, whether REPUBLICAN or DEMOCRAT

242) If they really wanted to vote Republican, they could go in the primary and say they were voting DEMOCRAT and then stack the ballot for someone that perhaps the REPUBLICAN could beat.

243) I do think more needs to be done along that line to help to, teach the everyone, more about what is going on with VOTING and with NONVOTING. So that, they 're making some more intelligent decisions.

244) I do n't know how it would be bringing in, CAT to a full grown DOG.

245) I would n't want to bring a CAT into a full grown DOG,

246) FIRING Tom Landry and HIRING Jimmy Johnson

Bibliography

Andersson, J., Cabral, J., Badino, L., Yamagishi, J., and Clark, R. (2009). Glottal source and prosodic prominence modelling in HMM-based speech synthesis for the Blizzard Challenge 2009. In *The Blizzard Challenge*, Edinburgh, U.K.

Badino, L., Andersson, J., Yamagishi, J., and Clark, R. (2009). Identification of contrast and its emphatic realization in HMM-based speech synthesis. In *Proc. Interspeech*, Brighton, U.K.

Badino, L. and Clark, R. (2007). Issues of optionality in pitch accent placement. In *Proc. 6th ISCA Speech Synthesis Workshop*, Bonn, Germany.

Baker, R., Clark, R., and White, M. (2004). Synthesising contextually appropriate intonation in limited domains. In *Proc. 5th ISCA workshop on speech synthesis*, Pittsburgh, USA.

Baum, L. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite-state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563.

Baumann, S. and Grice, M. (2006). The intonation of accessibility. *Journal of Pragmatics*, 38(10):1636–1657.

Beckman, M. and Pierrehumbert, J. (1986). Intonational structure in English and Japanese. *Phonology Yearbook*, 3:255–310.

Bolinger, D. (1958). A theory of pitch accent in English. *Word*, 14:109–149.

Bolinger, D. (1961). Contrastive accent and contrastive stress. *Language*, 37:83–96.

Bolinger, D. (1972). Accent is predictable (if you are a mind reader). *Language*, 49:633–644.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L., Friedman, J., Olshen, R., and Stone, P. (1984). *Classification and regression trees*. Wadsworth International Group, Belmont, CA, USA.

Brenier, J., Nenkova, A., Kothari, A., Whitton, L., Beaver, J., and Jurafsky, D. (2006). The (non) utility of linguistic features for predicting prominence in spontaneous speech. In *IEEE/ACL Workshop on Spoken Language Technology*, Aruba, USA.

Bruce, G. (1997). Swedish word accents in sentence perspective. Developing the Swedish intonation model, Working Papers, Department of Linguistics and Phonetics, University of Lund.

Bulyko, J. and M. Ostendorf, M. (2001). Joint prosody prediction and unit selection for concatenative speech synthesis. In *Proc. of ICASSP 2001*, Salt Lake City, USA,.

Calhoun, S. (2006). *Information Structure and the Prosodic Structure of English*. Phd thesis, University of Edinburgh, Edinburgh, UK.

Calhoun, S. (2008). Why do we accent words? the processing of focus and prosodic structure. In *Conference on Experimental and Theoretical Advances in Prosody*, Cornell University, NY, USA.

Calhoun, S., Nissim, S., Steedman, M., and Brenier, J. (2005). A framework for annotating information structure in discourse. In *Frontiers of Corpus Annotation II: pie in the Sky, ACL 2005 Conference Workshop*, Ann Arbor, Michigan, USA.

Campbell, N. and Beckman, M. (1997). Stress, prominence and spectral tilt. In *ESCA Workshop on Intonation: Theory, Models and Applications*, Athens, Greece.

Campillo, F. and Banga, R. (2006). A method for combining intonation modeling and speech unit selection in corpus-based speech synthesis systems. *Speech Communication*, 48:941–956.

Chafe, W. (1994). *Discourse, consciousness, and time: The flow and displacement of conscious experience in speaking and writing*. Chicago: University of Chicago Press, Chicago, USA.

Chen, K. and Hasegawa-Johnson, M. (2004). An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model. In *in Proc. of ICASSP*, Montreal, Quebec, Canada.

Chomsky, N. (1971). *Deep structure, surface structure, and semantic interpretation*. Semantics. Cambridge University Press., Cambridge, UK.

Chu, M., Zhao, Y., and Chang, E. (2006). Modeling stylized invariance and local variability of prosody in text-to-speech synthesis. *Speech Communication*, 48:716–726.

Clark, R. and King, S. (2006). Joint prosodic and segmental unit selection speech synthesis. In *Proc. Interspeech 2006*, Pittsburgh, USA.

Clark, R., Richmond, K., and King, S. (2007). Multisyn: Open-domain unit selection for the Festival speech synthesis system. *Speech Communication*, 49(4):317–330.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Proc. ESCA Eurospeech*, Rhodes, Greece.

Cohen, W. (1995). Fast effective rule induction. In *International Conference on Machine Learning*, Tahoe City, California, USA.

Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *42nd Annual Meeting of ACL*, Barcelona, Spain.

D. Jurafsky, D. and Martin, J. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Dasgupta, S. and Hsu, D. (2008). Hierarchical sampling for active learning. In *Twenty-Fifth International Conference on Machine Learning*, Helsinki, Finland.

Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Thirteenth International Joint Conference on Articial Intelligence*, Chambery, France.

Fellbaum, C. e. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Fery, C. and Samek-Lodovici (2006). Focus projection and prosodic prominence in nested foci. *Language*, 82(1):131–150.

Godfrey, J., Holliman, E., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, San Francisco, California, USA.

Gregory, M. and Altun, Y. (2004). Using conditional random fields to predict pitch accents in conversational speech. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain.

Grosz, B. and Sidner, C. (1996). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Guegan, M. and Hernandez, N. (2006). Recognizing textual parallelism with edit distance and similarity degree. In *11th Conference of EACL*, Trento, Italy.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

Halliday, M. (1967). Notes on transitivity and theme in English, part II. *Journal of Linguistics*, 3:199–244.

Hirschberg, J. (1993). Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence*, 63(1-2):305–340.

Hunt, A. and Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of the International Conference on Speech and Language Processing*, Philadelphia, PA, USA.

Joachims, T. (1999a). Making large-scale SVM learning practical. In Scholkopf, B., Burges, C., and (ed.), A. S., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

Joachims, T. (1999b). Transductive inference for text classification using Support Vector Machines. In *International Conference on Machine Learning*, Bled, Slovenia.

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA*, Tartu, Estonia.

Karaiskos, V., King, S., Clark, R., and Mayo, C. (2008). The Blizzard Challenge 2008. In *In Proc. Blizzard Challenge Workshop*, Brisbane, Australia.

Kiss, K. E. (1998). Identificational focus versus information focus. *Language*, 74(2):245–273.

Krahmer, E. and Swerts, M. (2001). On the alleged existence of contrastive accents. *Speech Communication*, 34(4):391–405.

Kruijff-Korbayova, I. and Steedman, M. (2003). Discourse and information structure. *Journal of Logic, Language and Information*, 12:249–259.

Kuhn, R. and De Mori, R. (1990). A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 12(6):570–583.

Ladd, D. R. (1996). *Intonational Phonology*. Cambridge University Press, Cambridge, UK.

Ladd, D. R. (2009). *Intonational Phonology, 2nd edition*. Cambridge University Press, Cambridge, UK.

Laffeerty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, Williamstown, MA, USA.

Levow, G. (2008). Automatic prosodic labeling with conditional random fields and rich acoustic features. In *IJCNLP*, Hyderabad, India.

Liberman, M. (1975). *The intonational system of English*. PhD thesis, MIT Linguistics, Cambridge, MA, USA.

Lin, D. (1998a). Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Lin, D. (1998b). An information-theoretic definition of similarity. In *Proc. of the International Conference on Machine Learning*, Madison, Wisconsin, USA.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Marsi, E. (2004). Optionality in evaluating prosody prediction. In *Proc. Of 5th ISCA Speech Synthesis Research Workshop*, Pittsburgh, USA.

Morik, K., Brockhausen, P., and Joachims, T. (1999). Combining statistical learning with a knowledge-based approach - A case study in intensive care monitoring. In *International Conference on Machine Learning*, Bled, Slovenia.

Needham, W. (1990). Semantic structure, information structure and intonation in discourse production. *Journal of Memory and Language*, 29(4):455–468.

Neeleman, A. and Szendroi, K. (2004). Superman sentences. *Linguistic Inquiry*, 35(1):149–159.

Nenkova, A., Brenier, J., Kothari, A., Calhoun, S., Whitton, L., Beaver, D., and Jurafsky, D. (2007). To memorize or to predict: Prominence labeling in conversational speech. In *Proceedings of NAACL*, Rochester, USA.

Nivre, J. (2006). w3.msi.vxu.se/ñivre/research/penn2malt.html.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kbler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Ostendorf, M., Price, P., and Shattuck-Hufnagel, S. (1995). The Boston University radio news corpus. Technical report, Electrical, Computer and Systems Engineering Department, Boston University.

Ostendorf, M., Shafran, I., Shattuck-Hufnagel, S., and Carmichael, L.and Byrne, W. (2001). A prosodically labeled database of spontaneous speech. In *Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, Red Banks, NJ, USA.

Pan, S. and Hirschberg, J. (2000). Modeling local context for pitch accent prediction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Hong Kong.

Pan, S. and McKeown, K. (1999). Word informativeness and automatic pitch accent modeling. In *Proc. of joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, University of Maryland, USA.

Pan, S., McKeown, K., and Hirschberg, J. (2002). Exploring features from natural language generation for prosody modeling. *Computer Speech and Language*, 16(3-4):457–490.

Patwardhan, S., Banerjee, S., and Pedersen, T. (2005). SenseRelate::Targetword - A generalized framework for word sense disambiguation. In *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). WordNet::Similarity - Measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, San Jose, CA, USA.

Phan, X., Nguyen, L., and C.T., N. (2005). FlexCRFs: Flexible Conditional Random Field Toolkit. http://flexcrfs.sourceforge.net/.

Pierrehumbert, J. (1980). *The Phonology and Phonetics of English Intonation*. PhD thesis, MIT, Cambridge, MA, USA.

Pitrelli, J., Beckman, M., and Hirschberg, J. (1994). Evaluation of prosodic transcription labelling reliability in the ToBI framework. In *Proceedings of the Third International Conference on Spoken Language Processing*, Yokohama, Japan.

Pitrelli, J. and Eide, E. (2003). Expressive speech synthesis using American English ToBI: Questions and contrastive emphasis. In *Proceedings of IEEE ASRU*, St. Thomas, Virgin Islands.

Postolache, O., Kruijff-Korbayova, I., and Kruijff, G. (2005). Data-driven approaches for information structure identification. In *Proceedings of HLT/EMNLP*, Vancouver, Canada.

Prevost, S. and Steedman, M. (1994). Specifying intonation from context for speech synthesis. *Speech Communication*, 15(1-2):139–153.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, USA.

Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In *Proc. of the Empirical Methods in natural Language Processing Conference*, University of Pennsylvania, USA.

Resnick, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*, Montreal, Canada.

Richards, O. and Baker, M. (2008). GridPP and the Edinburgh Compute and Data Facility or how a general purpose cluster bore the weight of atlas on its shoulders. In *Proc. UK All Hands Meeting*, UK.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62:107–136.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics*, 1:75–116.

Ross, K. and Ostendorf, M. (1996). Prediction of abstract prosodic labels for speech synthesis. *Computer Speech and Language*, 10(3):155–185.

Shannon, C. E. (1948). A mathematical theory of communication. *Bells System Technical Journal*, 27:379–423 and 623–656.

Silverman, K., Beckman, M., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). A standard for labelling English prosody. In *Proceedings of the International Conference on Spoken Language Processing*, Banff, Alberta, Canada.

Sproat, R. (1994). English noun-phrase accent prediction for Text-to-Speech. *Computer Speech and Language*, 8:79–94.

Sridhar, V. and Bangalore, S. Narayanan, S. (2008). Exploiting acoustic and syntactic features for automatic prosody labeling in a maximum entropy framework. *IEEE Transactions on Audio, Speech, and Language processing*, 16(4):797–811.

Sridhar, V., Nenkova, A., Narayanan, S., and Jurafsky, D. (2008). Detecting prominence in conversational speech: pitch accent, givenness and focus. In *4th Conference on Speech Prosody*, Campinas, Brazil.

Steedman, M. (2000). Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*, Denver, Colorado.

Strom, V., Clark, R., and King, S. (2006). Expressive prosody for unit-selection speech synthesis. In *Proc. Interspeech*, Pittsburgh, Pennsylvania, USA.

Strom, V., Nenkova, A., Clark, R., Vasquez-Alvarez, Y., Brenier, J., King, S., and Jurafsky, D. (2007). Modelling prominence and emphasis improves unit-selection synthesis. In *Proc. Interspeech*, Antwerp, Belgium.

Sun, X. (2002). Pitch accent prediction using ensemble machine learning. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, USA.

Sun, X. and Applebaum, T. (2001). Intonational phrase break prediction using decision tree and n-gramm model. In *Proc. of Eurospeech*, Aalborg, Denmark.

Sutton, C. and McCallum, A. (2007). An introduction to conditional random fields for relational learning. In Getoor, L. and Ben Taskar, e., editors, *Introduction to Statistical Relational Learning*. MIT Press.

Taylor, P. (2000). Analysis and synthesis of intonation using the Tilt model. *Journal of the Acoustical Society of America*, 107:1697–1714.

Taylor, P. (2009). *Text-to-Speech Synthesis*. Cambridge University Press, Cambridge, UK.

Taylor, P., Caley, R., Black, A., and King, S. (1999). Edinburgh Speech Tools Library. System Documentation Edition 1.2.

Terken, J. and Hirschberg, J. (1994). Deaccentuation of words representing given information: Effects of persistence of grammatical role and surface position. *Language and Speech*, 37:125–145.

Tokuda, K., Yoshimura, T., Masuko, T., and Kobayashi, T.and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, Istanbul, Turkey.

Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 27:45–66.

Turk, A. (1999). Structural influences on accentual lengthening in English. *Journal of Phonetics*, 27:171–206.

Umbach, C. (2004). On the notion of contrast in information structure and discourse structure. *Journal of Semantics*, 21:1–21.

Vallduví, E. and Vilkuna, M. (1998). One rheme and kontrast. *Syntax and Semantics*, 29:79–108.

Vapnik, V. (1982). *Estimation of dependencies based on empirical data*. Springer.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Vlachos, A. (2008). A stopping criterion for active learning. *Computer Speech and Language*, 22:295–312.

Weintraub, M. P. l. (1995). LM 95 project report. fast training and portability. Sri project report, SRI.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Witten, I. and Eibe, F. (2005). *Data Mining: Practical machine learning tools and techniques (2nd ed.)*. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Yuan, J., Brenier, J., and Jurafsky, D. (2005). Pitch accent prediction: Effects of genre and speaker. In *Proc. Interspeech*, Lisboa, Portugal.

Zen, H. and Toda, T. (2005). An overview of Nitech HMM-based speech synthesis system for Blizzard Challenge 2005. In *Proc. of Interspeech*, Lisboa, Portuga.

Zhang, T., Hasegawa-Johnson, M., and Levinson, S. (2006). Extraction of pragmatic and semantic salience from spontaneous spoken English. *Speech Communication*, 48(3-4):437–462.